Monetra®

Client Interface Protocol Specification

Programmer's Guide v7.2
Updated November 2009

Table of Contents

This page intentionally left blank.

1 Revision History

Date	*Rev.*	*Notes*
11/10/09	v7.2.0	List known addendums Sync missing features: - AVSOnly - Cardtype - NSF Flag - Date Formats - Msoft_code and phard_code updates - Encrypted export/import - RFC 4180 CSV response notes - Version request - Setlogging request - LicInfo request - Post-Settlement Batch Totals - Update new possible parameters for reports - Force Void request - Merchant Info request - CardLevel results - PINless debit - Prepaid gift card authamount and balance notes - Return by TTID - Ordernum and CustRef fields - Recurring/Installment Billing interchange fields - Account response parameter for ValueLink - Report CSV response updates
	v6.0.0	- mark batch=all and not specifying a sub account on settlement as a deprecated feature - bdate/edate – no date format limitation, if specified, both are required
	v5.5.0	- add missing keys 'origtype'/'voidorigtype' for toreversal
	v5.3.1	- rfid flag for proximity transactions
04/04/06	v5.3.0	- CHNGPWD request added on a user-administrative level - req_dial, req_ip, req_https, req_ssl, req_other, displayname, desc added to procfields request - displayname, helpdesk_phone added to proclist request

Date	*Rev.*	*Notes*
		- profile_id added to add/edit user requests - fieldedit clarifications - advancedeposit flag for lodging transactions - document 'NT' (nontaxable) value for tax param - Added 'VISADS' cardtype denoting debit transaction switched to credit (Visa only)
10/31/05	v5.2.1	- Toreversal clarification and missing 'timestamp' field. ACCOUNT_CLOSED phard_code added.
10/20/05	v5.2.0	- Add CAVV (verified by visa, mastercard secure code) flags. Add new msoft_code of DB_FAIL.
08/24/05	v5.1.2	- Missing TTID in general user requests parameters table. - Missing admintypes request parameter. - Missing apprcode in general user requests, and expanded on preauthcomplete/force description.
08/19/05	v5.1.1	- Added missing sub user information for the User Administrative Tasks - Added more documentation about sub users in the User Administration Appendix
08/01/05	v5.1	- Re-laid out protocol guide. - Synced functionality with Monetra 5.1. - Separated communications portion into "Monetra IP, SSL, and Dropfile Specification".

This page intentionally left blank.

2 Overview

2.1 Introduction to Monetra

Monetra (formerly MainStreet Credit Verification Engine) was designed from the ground up to support a variety of merchant environments such as Accounting, Point of Sale and web-based systems. From the very beginning, we have tried to incorporate the functionality of our clients' input directly into the delivered product. Monetra is constantly under development, and we are working hard to add more features to the core engine every day. If there are any features you would like to see included in Monetra, please contact engineering via support@monetra.com.

Due to the fact that Monetra is designed as a "Server" application, which can be integrated into a wide variety of applications (legacy and new), it is imperative that we provide a simple yet robust communication channel.

Every function the engine provides can be called/issued by the message formats below. These messages may be sent via DropFile, TCP/IP, SSL, XML-HTTP, XML-HTTPS or custom written communication modules.

You will need to reference other documentation for communication specifics located at http://www.monetra.com/content/developers.html .

2.2 Basic Protocol Description

Transactions sent to Monetra are made up of key/value pairs. The way these key/value pairs are sent to Monetra depends on the method of communication, please reference our IP, SSL, and DropFile Protocol Specification or XML Protocol Specification for more information on method-specific formatting.

Every transaction sent to Monetra is authenticated via a username and password combination (passed via the 'username' and 'password' keys), and has a transaction type (passed via the 'action' key).

Most transactions also accept a **`timeout`** key, which for attended (such as POS or E-Commerce) transactions is highly recommended to prevent an end-user from waiting too long. Monetra will always eventually hit an error condition and return a response if a transaction cannot be processed, but without a timeout hint being sent to Monetra, its priority will be to try to complete the transaction. Please reference the relevant sections for more information.

The system administrator's username is 'MADMIN', with a default password of 'password'. This password should be changed immediately upon the initial startup of Monetra.

User accounts may have 'subusers', which act as the master account to which they are tied, but are assigned their own password and permissions levels (including transaction types allowed, whether or not it is an attended or automated account, and if it is not a subuser of the MADMIN account, whether or not it has access to full account numbers). A subuser is assigned it's own username, but is always prefixed with the master account name separated by a colon **':'**. For example, if a sub account 'johndoe' was added to the MADMIN user, the username to login to Monetra would be 'MADMIN:johndoe'. You will want to reference A.1 User Administration for more information.

Response formats can either be 'standard' which refers to key/value pair responses, or 'comma delimited' which means it is returned in standard CSV (RFC4180) format. The CSV parser should be able to handle quoted responses (which may contain new line characters or commas) as outlined in RFC 4180.

2.3 How to Use This Guide

- Please note that each message format corresponds to a request and response parameters table listed in the following two sections.
- Each transaction type corresponds to a single 'Key Value', which is associated with the 'action' key. The 'Key Value' is located under each transaction description.
- For Administrative User Requests, the 'action' key is always set to a value of 'admin', and you should set the 'admin' key to the 'Admin Value' as documented under each transaction description.
- In each parameters table, you will find the following headings:
  - **Key**- denotes the key of the key/value pair
  - **Req**- states whether a param key is required, optional, or conditionally required.
  - **Tran Type**- denotes transaction type for which the parameter is valid. May be a negative reference (prefixed by minus sign, which indicates everything except that transaction type)
  - **Ref**- Additional references for this field
  - **Description**- Detailed description of field, also describes conditional nature
- When dealing with user accounts, you MUST reference the Required Processor Specific Information as it appears in each parameters table. It is critical that the user accounts are added 100% properly before the account goes into a live environment.
- The general transaction response fields are listed in a separate table along with a corresponding result code appendix which lists the codes themselves and their descriptions. The following response fields have appendices: code, msoft_code, phard_code, avs, and cv.
- Cross reference when necessary to retrieve relevant information.

2.4 Key Descriptions

Requirement Command	*Description*
Opt	Optional Parameter
Y	Required for transaction
C	Conditionally Required in some circumstances

2.5 Addendums

There are known addendums for:
- Check Processing
- Monetra DSS Secure Card storage and Billing
- Interchange reporting
- Bill Me Later

These should be at http://www.monetra.com/content/developers.html. If you cannot find the addendum you are looking for, please contact support@monetra.com.

3 Transaction Types

3.1 Engine Administrative Requests

3.1.1 Add Monetra User

Key Value: **adduser**
Description: adds user to Monetra database (tied to merchant account)
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.1 User Administration

3.1.2 Check Password

Key Value: **chkpwd**
Description: checks the validity of a password
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.3 Change Password

Key Value: **chngpwd**
Description: changes MADMIN user's password
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.4 Clear Connection Log

Key Value: **clearconnlog**
Description: clears raw connection data from DB
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.5 Clear Communications Error Log

Key Value: **`clearerrorlog`**
Description: clears raw communications errors from DB
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.6 Clear Transaction Details Log

Key Value: **`cleartranlog`**
Description: clears raw transaction data from DB
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.7 List Connection Details

Key Value: **`connlog`**
Description: lists raw connection data on a per-engine basis
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: comma delimited
Response Params: *`connid,starttimestamp,endtimestamp,ipaddr,method,closemsg`*

3.1.8 Automated Tasks (CRON)

Key Value: **`cron`**
Description: allows engine admin to schedule tasks to run automatically
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard or comma-delimited (if **`cron=list`**)
Response Params: *`cronid,cron_task,cron_date,last_ts,next_ts`*
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.3 Automated Task (CRON) Management

3.1.9 Delete Monetra User

Key Value: **`deluser`**
Description: deletes a user from the Monetra database
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.1 User Administration

3.1.10 Disable Monetra User

Key Value: **`disableuser`**
Description: temporarily disables user without deleting
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.1 User Administration

3.1.11 Edit Monetra User

Key Value: **`edituser`**
Description: edits user information in Monetra database
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.1 User Administration

3.1.12 Enable Monetra User

Key Value: **`enableuser`**
Description: re-enables previously disabled users
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.1 User Administration

3.1.13 List Communications Errors

Key Value: **`errorlog`**
Description: lists raw communication errors on per-engine basis
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: comma delimited
Response Params: *`marker,timestamp,devicetype,proc,conninfo,errcode,verbiage`*

3.1.14 Export Monetra Database

Key Value: **`export`**
Description: exports existing Monetra database to a file
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.15 Get Permissions

Key Value:	**`getperms`**
Description:	gets permission for functionality allowed by this user/subuser
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Params:	*`trantypes,unattended`*
Reference:	A.1 User Administration

3.1.16 Get Sub Accounts

Key Value:	**`getsubaccts`**
Description:	retrieves sub accounts for user (for split routing)
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Param:	*`sub`*
Reference:	A.1 User Administration

3.1.17 List Monetra User Information

Key Value:	**`getuserinfo`**
Description:	Lists all user merchant parameters as entered into Monetra database. Please reference the user appendix for information on additional parameters that may be returned that are not listed in the response table
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	standard
Response Table:	3.3 Engine Administrative Requests Response Table (pg 27)
Reference:	A.1 User Administration

3.1.18 Import Monetra Database

Key Value:	**`import`**
Description:	loads data from local file to Monetra database
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	standard
Response Table:	3.3 Engine Administrative Requests Response Table (pg 27)

3.1.19 License Info

Key Value:	**`licinfo`**
Description:	Retrieves private license information such as transaction limits, company name, etc.
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	standard
Response Table:	3.3 Engine Administrative Requests Response Table (pg 27)

3.1.20 List Engine Statistics

Key Value:	**`liststats`**
Description:	lists completed/settled transaction statistics on users
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Params:	*`user,totaltransNum,totaltransAmount,totalAuthNum, totalAuthAmount,totalReturnNum,totalReturnAmount`*

3.1.21 List Monetra Users

Key Value:	**`listusers`**
Description:	lists out users in Monetra database and shows enabled/disabled status
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Params:	*`userlist,user status,master,trantypes,admintypes, profile_id`*
Reference:	A.1 User Administration

3.1.22 Set System Maintenance Mode

Key Value:	**`maintenance`**
Description:	puts the engine into one of three maintenance models
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	standard
Response Table:	3.3 Engine Administrative Requests Response Table (pg 27)

3.1.23 Processor Fields

Key Value:	**`procfields`**
Description:	retrieves list of eligible parameters for merchant setup
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Params:	*`name,req_credit,req_debit,req_gift,req_ebt,req_dial, req_ip,req_https,req_ssl,req_other,len_min,len_max, field_type,displayname,desc`*

3.1.24 Processor List

Key Value:	**`proclist`**
Description:	returns list of processors, whether or not they are active, and capabilities of the processor
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Params:	*`proc,active,cardtypes,credit_trantypes,debit_trantypes, ebt_trantypes,gift_trantypes,check_trantype, supported_conns,supported_modes,displayname,helpdesk_phone, version,features`*

3.1.25 Show Processor Status

Key Value:	**`procstatus`**
Description:	allows engine admin to query engine for particular processors connection status
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	standard
Response Table:	3.3 Engine Administrative Requests Response Table (pg 27)

3.1.26 Restart Engine

Key Value: **`restart`**
Description: tells the Monetra engine to restart
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.27 Restrictions for Account Access

Key Value: **`restriction`**
Description: adds limitations to login access
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: comma delimited or standard (if **`restriction=list`**)
Response Params: *`id,type,data,len`*
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.2 Login Restriction Management

3.1.28 Set Logging Level

Key Value: **`setlogging`**
Description: sets the current logging level for Monetra.
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.29 Shutown Engine

Key Value: **`shutdown`**
Description: tells the Monetra engine to shut down
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)

3.1.30 Add Subuser (MADMIN)

Key Value: **`subuseradd`**
Description: allows engine admin to add subuser for administrative tasks
Access Level: Administrator
Param Table: 3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type: standard
Response Table: 3.3 Engine Administrative Requests Response Table (pg 27)
Reference: A.1 User Administration

3.1.31 Delete Subuser (MADMIN)

Key Value: **`subuserdel`**
Description: allows engine admin to delete a subuser account
Access Level: Administrator

Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	standard
Response Table:	3.3 Engine Administrative Requests Response Table (pg 27)
Reference:	A.1 User Administration

3.1.32 Edit Subusers (MADMIN)

Key Value:	**`subuseredit`**
Description:	allows engine admin to edit a subuser accounts
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	standard
Response Table:	3.3 Engine Administrative Requests Response Table (pg 27)
Reference:	A.1 User Administration

3.1.33 List Subusers (MADMIN)

Key Value:	**`subuserlist`**
Description:	allows engine admin to list subuser accounts for MADMIN
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Table:	*`user,pwd,master,trantypes,admintypes,obscure,unattended`*
Reference:	A.1 User Administration

3.1.34 List Transaction Details

Key Value:	**`tranlog`**
Description:	lists auditable transaction details with connection identifier matching on a per-engine basis
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Params:	*`connid,starttimestamp,endtimestamp,trantype,admintype, user,ttid,result`*

3.1.35 Version

Key Value:	**`version`**
Description:	lists version of Monetra as first row, remaining rows contain a list of all loaded modules and their versions
Access Level:	Administrator
Param Table:	3.2 Engine Administrative Requests Parameters Table (pg 23)
Response Type:	comma delimited
Response Params:	*`name,displayname,type,version`*

3.2 Engine Administrative Requests Parameters Table

Key	*Req*	*Tran Type*	*Ref*	*Description*
username	Y	ALL		always MADMIN or subuser of MADMIN
password	Y	ALL		password associated with user
action	Y	ALL		action as referenced by 'Key Value' in previous section
annual_trans_limit	Y	licinfo		Annual Transaction Limit
bdate	C	connlog, clearconnlog, tranlog, cleartranlog	A.13 Date Formats	begin date. Required if edate is specified.
cardtypes	C	adduser, enableuser	A.10 Card Types	card types desired to support, delimited by pipes or pluses ('|' or '+')
compname	Y	licinfo		Company name who owns the license
connid	C	connlog, clearconnlog, tranlog, cleartranlog		unique identifier for connection as returned from audit
cron	Y	cron	A.3 Automated Task (CRON) Management	Specify the cron function to perform. '**add**', '**list**', or '**remove**' are valid options.
cron_date	C	cron	A.3 Automated Task (CRON) Management	Date for cron formatted as described in appendix
cronid	C	cron	A.3 Automated Task (CRON) Management	Unique cron identifier as returned from audit
cron_task	C	cron	A.3 Automated Task (CRON) Management	the cron tasks available to Engine Administrators are limited to '**DBMAINT**', and custom functions. Please review the cron_task reference.
debug	Y	setlogging	A.14 Debug Levels	Bitmapped values separated by | or + . Explicitly set the desired logging level for the current Monetra instance. The information set here will be lost upon restart of Monetra. Using the setlogging function is the only way to increase the log level to

Key	Req	Tran Type	Ref	Description
				non-PCI compliant levels.
devicetype	C	errorlog, clearerrorlog		DIAL, HTTPS, IP, SSL, or OTHER. Audit for error based on connectivity method.
edate	C	connlog, clearerrorlog, tranlog, cleartranlog	A.13 Date Formats	end date. Required if bdate is specified.
email	Opt	adduser, enableuser		email address for sending notices/CRN emails
encrypt	Opt	export		Boolean. If set to true, will create an encrypted export and return the 'key' response parameter to use for import.
errorcode	C	errorlog, clearerrorlog		TIMEOUT, RDISCONNECT, CONNFAILED, NOENQ, DEVERROR, FAILOVER Audit for error based on type of connectivity failure.
exp_end	C	licinfo		Demo license expiration date
exp_runtime	C	licinfo		Number of seconds a demo may run before expiring.
exp_start	C	licinfo		Demo license begin date
file	Y	export, import		absolute path to file for import/export
indcode	Y	adduser, enableuser	A.12 Industry Codes	Industry code for processing transactions (See appendix)
ipaddr	C	connlog		ip address to audit connection log
key	C	import		Required if encrypted export was performed. This should be the unaltered key in the response to the export request.
level	Y	maintenance		'**strict**', '**limited**', or '**none**' strict disallows any commands besides maintenance, import, and export. Limited disables add/edit/delete user requests, and settlements, but allows standard transactions. None is default and opens up everything.
license_id	Y	licinfo		Monetra license id
localhost_only	Y	licinfo		If connections to Monetra are only allowed on localhost or not
marker	C	errorlog		Unique error marker for communications error as seen in logs

Key	Req	Tran Type	Ref	Description
max_version	Y	licinfo		Maximum Monetra version this license supports
merchant_restrictions	Y	licinfo		Any restrictions which may exist on what processing institution or merchant account numbers may be used.
method	C	connlog		IP/SSL, DropFile, XML. Audit for the incoming connection type to Monetra
mode	C	adduser, edituser		A merchant account can be set up for AUTH, SETTLE, or BOTH. This allows addition of sub accounts, where one authorizes the transaction, and the other settles.
num_users		licinfo		Number of user/merchant accounts allowed.
os	Y	licinfo		The OS the license is provisioned for.
partial	Opt	export		SETUP, MISC, or DATA. SETUP will export the users and merchant account information. MISC will export the ttid, batch, and item numbers. DATA will export the transaction history (unsettled, failed, and settled).
partner_id		licinfo		Partner ID who sold the license
proc	Y	adduser, edituser, procstatus, clearerrorlog, errorlog, procfields	A.1 User Administration	Processing institution supported by Monetra's current configuration.
profile_id	N	adduser,edituser	A.1 User Administration	Profile id as stored on Main Streets servers referencing the merchant account in the system, for use with the Wizard.
pwd	C	adduser, edituser, subuseradd, subuseredit, chngpwd		password to assign merchant
req_signed_mods	Y	licinfo		Whether or not signed modules are required
required_modules	Y	licinfo		semi-colon separated list of modules which must be loaded.
restriction	Y	restriction	A.2 Login Restriction Management	'**add**', '**list**', or '**remove**' sub-function may be specified here

Key	Req	Tran Type	Ref	Description
restriction_data	Y	restriction	A.2 Login Restriction Management	hexidecimal fingerprint data (each byte separated by colons)
restriction_type	Y	restriction	A.2 Login Restriction Management	currently only valid type is '**ssl_cert**'
sqlite_only	Y	licinfo		Whether or not the license allows databases besides SQLite
sub	C	adduser, edituser, deluser, getuserinfo		integer value for sub accounts. 0 is reserved for primary account, and must exist. If deleting sub 0, entire user account will be deleted. May use arbitrary numbers above this to assign sub accounts
trantypes	Y	subuseredit, subuseradd		list of transaction types you wish to allow sub user to utilize. This list should be delimited by pipes or pluses ('|' or '+')
ttid	C	tranlog		unique identifier for a transaction per merchant account
user	Y	adduser, deluser, edituser, enableuser, disableuser, getuserinfo, subuserlist, tranlog, subuseradd, subuseredit, subuserdel, getsubaccts, restriction		user account editing/adding/auditing

3.3 Engine Administrative Requests Response Table

Key	*Tran Type*	*Ref*	*Description*
code	ALL	A.4 Result Codes (code)	Result code for transaction describing success or failure
msoft_code	ALL	A.5 Result Codes (msoft_code)	If an error condition occurred, this is a more detailed result code generated by Monetra
verbiage	ALL		Human-Readable result, meant to be passed-on to display
cardtypes	getuserinfo	A.1 User Administration	
mode	getuserinfo	A.1 User Administration	
conn_priority	getuserinfo	A.1 User Administration	
email	getuserinfo	A.1 User Administration	
configuredmethods	procstatus		semi-colon delimited list of configured connection methods for processor, ordered by priority, may be a combination of ssl;https;ip;dial;other
currentmethod	procstatus		the current default connectivity method to use for the processor, takes into account offline methods
consecerrors	procstatus		Consecutive error (connectivity) count for processor/method
onlinemethods	procstatus		semi-colon delimited list of online connectivity methods. Same formatting/parameters allowed as 'configuredmethods'
offlinemethods	procstatus		semi-colon delimited list of offline connectivity methods, which each internal value is delimited by a colon with the first value being the method, and the second value being how long the method has been offline in seconds. (Ex. ssl:60;https:30)

3.4 General User Requests

3.4.1 Activate

Key Value:	**`activate`**
Description:	activates a stored value card with the processor.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.2 AVS Only

Key Value:	**`avsonly`**
Description:	Performs a verification on AVS and CV data, should be used instead of performing $1.00 preauths. (aka Verification Only)
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.3 Balance Inquiry

Key Value:	**`balanceinq`**
Description:	retrieves the remaining balance on card/account.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.4 Capture Transaction

Key Value:	**`capture`**
Description:	adds transaction to settlement queue (if original authorization's capture flag was set to no). Not allowed on preauth transactions.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.5 Card Type

Key Value:	**`cardtype`**
Description:	Performs a card type looking using Monetra's BIN table. Useful for POS systems who do not maintain their own BIN table.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.6 Cash Out

Key Value:	**`cashout`**
Description:	retrieves all remaining funds on card
Access Level:	User
Card Types:	Gift
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.7 EBT Cash Benefits Balance

Key Value:	**`ebtcbbalance`**
Description:	balances inquiry transaction for Cash Benefits
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.8 EBT Cash Benefits Cash Withdrawal

Key Value: **ebtcbcash**
Description: similar to EBT Sale, except full amount is cashback
Access Level: User
Param Table: 3.5 General User Request Parameters Table (pg 34)
Response Type: standard
Response Table: 3.6 General User Requests Response Table (pg 41)

3.4.9 EBT Cash Benefits Sale

Key Value: **ebtcbsale**
Description: EBT Sale for Cash Benefits
Access Level: User
Param Table: 3.5 General User Request Parameters Table (pg 34)
Response Type: standard
Response Table: 3.6 General User Requests Response Table (pg 41)

3.4.10 EBT Food Stamps Balance

Key Value: **ebtfsbalance**
Description: EBT Balance Inquiry transaction for Food Stamps
Access Level: User
Param Table: 3.5 General User Request Parameters Table (pg 34)
Response Type: standard
Response Table: 3.6 General User Requests Response Table (pg 41)

3.4.11 EBT Food Stamps Return

Key Value: **ebtfsreturn**
Description: EBT Return for Food Stamps
Access Level: User
Param Table: 3.5 General User Request Parameters Table (pg 34)
Response Type: standard
Response Table: 3.6 General User Requests Response Table (pg 41)

3.4.12 EBT Food Stamps Sale

Key Value: **ebtfssale**
Description: EBT Sale for Food Stamps
Access Level: User
Param Table: 3.5 General User Request Parameters Table (pg 34)
Response Type: standard
Response Table: 3.6 General User Requests Response Table (pg 41)

3.4.13 EBT Food Stamps Voucher

Key Value: **ebtfsvoucher**
Description: phone Authorization data entry for Food Stamps
Access Level: User
Param Table: 3.5 General User Request Parameters Table (pg 34)
Response Type: standard
Response Table: 3.6 General User Requests Response Table (pg 41)

3.4.14 Force/PreauthComplete

Key Value:	**force**, **preauthcomplete**
Description:	Completes an authorized transaction (adds it to the settlement batch). May be a phone authorization or a preauth run through the software. There are two ways to run this transaction type. One requires the ptrannum or ttid of the original transaction if processed as a preauth through Monetra, as well as the final amount of the transaction. The other requires the full transaction data including the account number, expiration date, amount, and approval code.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.15 Incremental

Key Value:	i**ncremental**
Description:	on visa cards, this function will attempt to add funds onto a previous authorization by verifying with processing institution.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.16 Issue

Key Value:	**issue**
Description:	essentially the same as activate, though some gift processors support both.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.17 Interactive Voice Request

Key Value:	**ivrreq**
Description:	phone authorization for gift cards (Interactive Voice Response)
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.18 Interactive Voice Response

Key Value:	**ivrresp**
Description:	Passes result received from IVR system
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.19 Merchant Return

Key Value:	**`merchreturn`**
Description:	similar to a return/reload, except the back end of the processor will denote it as a merchandise return.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.20 Preauth

Key Value:	**`preauth`**
Description:	functionally equivalent to running a SALE with capture set to no.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.21 Return/Reload/Credit

Key Value:	**`return`**, **`reload`**, **`credit`**
Description:	On credit cards, this is used post-settlement to refund a purchase, but no prior history is necessarily required. On debit cards, a previous transaction must be referenced, and is used for refunds. On Gift cards, this may be used for refunds, or to add money to a card.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.22 Reversal

Key Value:	**`reversal`**
Description:	real-time removal of funds on VISA cards. If amount is equal to the full transaction amount, the end result is similar to void. Partial amounts allowed.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.23 Sale/Redemption

Key Value:	**`sale`**, **`redemption`**
Description:	function debits available funds from cardholder account.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.24 Settle

Key Value:	**settle**
Description:	transactions at the end of the day must be 'settled' which will tell the funds to deposit into the account.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.25 Settle Request for Response

Key Value:	**settlerfr**
Description:	some big-batch processors require, when a status of a settlement is unknown (because of disconnect), than a Request for Response is sent as an inquiry. If a response is located on the host, it will be processed on Monetra's end, just as if a settlement was issued. This typically will only be used when a processor tells you to issue such a command.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.26 Timeout Reversal

Key Value:	**toreversal**
Description:	on GIFT cards, it may be necessary to issue a TOREVERSAL if Monetra responds saying to do so (msoft_code of CONN_TOREVERSAL). This is because GIFT cards are real-time, and if a disconnect occurs in the middle of a transaction, Monetra has no means to know if the processor had successfully completed the operation or not. You should send all the original transaction details back, but set 'action' to 'toreversal' and 'origtype' to your original 'action'. It is also necessary to send the 'timestamp' that was returned from the Monetra response indicating a TOREVERSAL was necessary
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.4.27 Void

Key Value:	**void**
Description:	Removes a transaction from the settlement batch. May be online or offline.
Access Level:	User
Param Table:	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.6 General User Requests Response Table (pg 41)

3.5 General User Request Parameters Table

Key	Req	Acct Type	Tran Type	Ref	Description
username	Y	ALL	all		General user (or subuser of a general user) username. Must NOT be 'MADMIN'
password	Y	ALL	all		password associated with user/subuser
action	Y	ALL	all		appropriate 'Key Value' for General User Request.
account	C	ALL	all, -capture, -reversal, -incremental, -void, -preauthcomplete		If trackdata not present, the account number must be present. Should not be specified if referencing a ttid.
advancedeposit	Opt	Credit	sale,preauth,force		Field is used for advance deposits on Lodging transactions. This is required if using Advance Deposit features. Defaults to 'no', accepts 'yes' or 'no'.
amount	Y	ALL	all, -capture, -void, -cashout, -balanceinq, -ebtcbbalance, -ebtfsbalance, -cardtype, -avsonly		Amount of transaction. All amounts are positive. This should be an aggregate amount (e.g. already includes tax and examount)
apprcode	C	Credit	force/preauthcomplete		If a ttid/ptrannum is not supplied, this field is the approval code (aka authorization number) that the processing institution returned for the authorization, whether from another device or a phone auth.
batch	Y	ALL	settle, settlerfr		Batch number for settlement. 'ALL' is a deprecated feature which cause Monetra to settle any open batches in the background.
bdate	C	Credit	sale, preauth, force/preauthcomplete	A.13 Date Formats	Start date of visit for lodging. Required if edate is specified

Key	Req	Acct Type	Tran Type	Ref	Description
capture	Opt	Credit	sale, force/preauthcomplete, return,reload,credit		If capture is set to 'no', the transaction will not be added to the batch settlement. This value defaults to 'yes'.
cardholdername	Opt	Credit Debit EBT	all, -capture, -void, -cashout, -balanceinq, -ebtcbbalance, -ebtfsbalance		If track1 data (or combined track1/track2 data) is sent, this field is automatically populated, otherwise this field may be specified for reporting
cardpresent	Opt	Credit	sale, preauth, force/preauthcomplete		On retail/restaurant accounts where the card is not present at the time of the transaction, this flag should be set to 'no', and it will be processed in a similar fashion to MailOrder/PhoneOrder. Ignored for anything besides Retail or Restaurant accounts.
cashbackamount	Opt	Debit EBT	sale, ebtcbsale		Amount of funds requested is for cash to be given to the client. Required if customer is receiving cash
cavv	Opt	Credit	sale,preauth		Set to 'NONPARTICIPANT' if issuer is a non-participant, issuer is a participant but cardholder is not, or the authentication server is not available. Otherwise should pass the up to 40 character base64-encoded CAVV response from the VISA/MC authentication servers. This field is used for Verified by Visa and Mastercard Secure Code transactions.
clerkid	Opt	ALL	all		25 character user-defined reporting field
comments	Opt	ALL	all		50 character user-defined reporting field
curr	C	Credit	sale, preauth,		If processing in a foreign

Key	Req	Acct Type	Tran Type	Ref	Description
			force/preauthcomplete		currency, depending on the processor, this may be required
custref	Opt	Credit	sale,preauth		Alpha-numeric. This is a non-indexed field, so this cannot be queried, but will be sent on to the processing institution for Level II interchange qualification. This can be used to specify a customer PO number or other customer reference number.
cv	Opt	Credit	sale, preauth		Card Verification value. Usually 3 digits on back of VISA/MC/Discover, or 4 digits on front of AMEX cards. Used to help with fraud prevention. It's use is recommended for MOPO and Ecommerce industries.
dentalamount	Opt	Credit	sale, preauth		If healthcare=true, this may be specified to denote how much of the transaction is for dental.
descloc	Opt	Credit	sale, preauth, force/preauthcomplete		Merchant descriptor location. Formatting varies from processor to processor. (only SALEM right now)
descmerch	Opt	Credit	sale, preauth, force/preauthcomplete		Merchant descriptor name. Formatting varies from processor to processor. (only SALEM right now)
edate	C	Credit	sale, preauth, force/preauthcomplete	A.13 Date Formats	End date of visit for Lodging. Required if bdate is specified.
examount	Opt	Credit	sale, preauth, force/preauthcomplete		For restaurant, this is the tip amount.
excharges	Opt	Credit	sale, preauth, force/preauthcomplete	A.11 Excharges Values	For lodging, this defines the available additional charges that may have been in addition to the room rate.

Key	Req	Acct Type	Tran Type	Ref	Description
expdate	C	Credit Debit EBT	all, -capture, -reversal, -incremental, -void, -preauthcomplete		If trackdata not present, and this is not a gift card, this field must be present . Should not be specified if referencing a ttid.
healthcare	C	Credit	sale, preauth		Boolean. If the transaction is attempting to qualify for IIAS/FSA/HRA, this should be set to true.
installment_num	C	Credit	sale, preauth		Required if recurring=installment. This is the current payment number.
installment_total	C	Credit	sale, preauth		Required if recurring=installment. This is the total number of installment payments.
issuenum	C	Credit	sale/redemption, preauth, force/preauthcomplete		switch/solo cards require this field if startdate is not present
ksn	C	Debit EBT	sale, return, reload,credit, ebtcbsale, ebtcbcash, ebtcbbalance, ebtfssale, ebtfssale,ebtfsreturn, ebtfsvoucher, ebtfsbalance		required on pin-based debit, ebt requires it if doing pin-based transactions. (aka Key Serial Number) Exception: For PINless Debit bill payments, this would not be specified if pin=pinless.
ordernum	Opt	ALL	all		Like ptrannum, but is alpha-numeric. This is a non-indexed field, so this cannot be queried, but will be sent on to the processing institution for interchange qualification. The numeric portions of this will be copied into the ptrannum field if ptrannum is not specified.
origtype	Y	ALL	toreversal		Origtype should be specified as the original 'action' key used for the transaction for which a timeout reversal is needed.
otheramount	Opt	Credit	preauth, sale		If healthcare=true, this

Key	Req	Acct Type	Tran Type	Ref	Description
					may be specified to denote how much of the transaction is for other qualified purchases (such as clinic expenses).
nsf	N	Credit Gift	preauth, sale/redemption		Boolean. If true, will attempt to perform a partial authorization if insufficient funds remain on card.
pin	C	Debit EBT Gift	sale, return,reload,credit, ebtcbsale, ebtcbcash, ebtcbbalance, ebtfssale, ebtfsreturn, ebtfsvoucher, ebtfsbalance		required on pin-based debit, ebt requires it if doing pin-based. Some gift cards processors may want a pin for e-commerce transactions. For PINless Debit bill payments, this should be set to 'pinless'.
priority	Opt	ALL	all		low, normal, high (you should process large batches as priority low, so that any real-time transactions that come through will be processed immediately) defaults to normal.
ptrannum	C	ALL	all		Some processing institutions may require this field, which gets passed on as the order number. Numeric-only.
rate	C	Credit	sale, preauth, force/preauthcomplete		Room rate per night. Required on lodging.
recurring	Opt	Credit	sale, preauth		Takes values: **Yes**: recurring **No**: not recurring (default) **First**: First recurring payment **Installment**: installment payment (requires installment_num and installment_total params)
rfid	Opt	Credit/Debit	sale,preauth,force		If transaction was accepted via RFID

Key	Req	Acct Type	Tran Type	Ref	Description
					(proximity), this flag should be set to 'yes', and the trackdata parameter must also be sent. Some processors require track2 data, and track1 alone is not sufficient in that instance.
rxamount	Opt	Credit	sale,preauth		If healthcare=true, this may be specified to denote how much of the transaction is for prescriptions.
startdate	C	Credit	sale, preauth, force/preauthcomplete		switch/solo cards require this field if issuenum is not present
stationid	Opt	ALL	all		25-character user-defined reporting field
street	Opt	Credit	sale, preauth, avsonly		street address for AVS. Typically only numbers need to be passed in this fields. Letters and other characters are ignored.
sub	Y	ALL	settle, settlerfr		Sub account you wish to settle, assuming you have more than 1 for split-routing. Not specifying a sub account is a deprecated feature.
tax	Opt	ALL	sale, preauth, force/preauthcomplete		For purchase card transactions, this field is required to prevent 'downgrading'. It is the taxable amount on the purchase. For non-taxable transactions (tax exempt) specify 'NT' (without the quotes).
timeout	Opt	ALL	all		value in seconds that the transaction can sit in the Monetra queue without being timed-out. A transaction is 'locked' and will not be timed-out while being processed, can only time-out while in idle state.
timestamp	Y	Gift	toreversal		The timestamp returned

Key	Req	Acct Type	Tran Type	Ref	Description
					from the original transaction response, stating you must perform a toreversal.
trackdata	C	ALL	all, -capture, -reversal, -incremental, -void, -preauthcomplete		If account is not present, this must be present. May be track1, track2, or a combined track1/track2 read. Typically recommended to pass the data to Monetra exactly as received (track1/track2 combined). Should not be specified if referencing a ttid.
ttid	C	ALL	void, incremental, reversal, preauthcomplete, capture, return		This is the unique identifier Monetra returns with every transaction. It should be the preferred identifier to use to reference a transaction. Alternatively, you can use the user-definable ptrannum if kept unique (except on Return by TTID).
visionamount	Opt	Credit	sale, preauth		If healthcare=true, this may be specified to denote how much of the transaction is for vision.
voidorigtype	C	ALL	toreversal		If issuing a timeout reversal for a void, you 'may' need to specify the type of transaciton being voided.
zip	Opt	Credit	sale, preauth, avsonly		zipcode for AVS verification. All non-trackdata transactions require this to avoid being 'downgraded'.

3.6 General User Requests Response Table

Key	*Acct Type*	*Tran Type*	*Ref*	*Description*
account	Gift	sale, preauth, activate, issue, cashout, merchreturn, cardtype, balanceinq		If trackdata is specified and the printed card number is different from the cardnumber in the trackdata, it will return the cardnumber as printed on the card for verification. (Primarily for ValueLink)
auth	ALL	sale, preauth, activate, issue, cashout, merchreturn, toreversal, ebtfssale, ebtfsreturn, ebtcbsale, ebtcbcash		authorization number, typically 6 digits numeric-only, but some processors in test mode will return alpha-numeric responses
authamount	Credit Gift	sale		amount actually authorized (for partial auths). Applies to Private Label Gift as well as Branded [Visa|MC|Amex] gift cards or Health Cards.
avs	Credit	sale, preauth	A.7 Result Codes (AVS)	address verification result
balance	Gift Debit EBT Credit	sale, balanceinq, ebtfsbalance, ebtcbbalance		balance remaining on card. On Credit, only returned if the card is a prepaid card or Health Card.
batch	ALL	all, - void, -settle, -settlerfr		batch number to which transaction was assigned
batconnum	ALL	settle, settlerfr		some processors return an authorization number-like element when you settle a batch
cardlevel	Credit	preauth, sale, avsonly	A.9 Visa Card Level Results	This is Monetra's interpretation of the visa 62.23 card level result returned by most processing institutions.
code	ALL	all	A.4 Result Codes (code)	result code of transaction
cv	Credit	sale, preauth	A.8 Result Codes (CV)	Card Verification result code
item	ALL	all, -void, -settle, -settlerfr		item number to which transaction was assigned
msoft_code	ALL	all	A.5 Result	detailed result code specific

Key	*Acct Type*	*Tran Type*	*Ref*	*Description*
				to Monetra-internal checks
pclevel	Credit	sale, preauth		whether or not it was a Purchase card transaction- 0=no, 1=business card, 2=purchase card
phard_code	ALL	sale, preauth, activate, balanceinq, cashout, toreversal, merchreturn, return (debit/gift only), issue, ebtfs*, settle, settlerfr	A.6 Result Codes (phard_code)	detailed result code for success/fail from processor
raw_avs	Credit	sale, preauth		raw code from processor for avs result
raw_code	ALL	all		raw code from processor on result code
raw_cv	Credit	sale, preauth		raw code from processor for cv result
raw_cardlevel	Credit	sale,preauth,avsonly	A.9 Visa Card Level Results	raw cod from processor for visa 62.23 cardlevel result
stan	Debit EBT	sale, ebtfssale, ebtcbsale		system trace audit number for transaction on processor end
timestamp	ALL	all		Unix timestamp of transaction (seconds since January 1, 1970), used mainly for TOReversals.
ttid	ALL	all		transaction id guaranteed to be unique across all transactions for a particular merchant
verbiage	ALL	all		textual, human-interpretable response code (meant for system display to clerk)

3.7 Administrative User Requests

3.7.1 Batch Totals

Key Value: ADMIN
Admin Value: **BT**
Description: returns batch totals for each unsettled batch
Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: comma delimited
Response Params: *BatchNum,sub,totaltransNum,totaltransAmount,totalAuthNum, totalAuthAmount,ReturnNum,totalReturnAmount,NumVisaAuth, AmntVisaAuth,NumVisaReturn,AmntVisaReturn,NumMCAuth, AmntMCAuth,NumMCReturn,AmntMCReturn,NumDiscAuth, AmntDiscAuth,NumDiscReturn,AmntDiscReturn,NumAmexAuth, AmntAmexAuth,NumAmexReturn,AmntAmexReturn,NumDinersAuth, AmntDinersAuth,NumDinersReturn,AmntDinersReturn,NumCBAuth, AmntCBAuth,NumCBReturn,AmntCBReturn,NumJCBAuth,AmntJCBAuth, NumJCBReturn,AmntJCBReturn,NumGIFTAuth,AmntGIFTAuth, NumGIFTReturn,AmntGIFTReturn,NumOtherAuth,AmntOtherAuth, NumOtherReturn,AmntOtherReturn,NumDebitAuth,AmntDebitAuth, NumDebitReturn,AmntDebitReturn,NumEBTAuth,AmntEBTAuth, NumEBTReturn,AmntEBTReturn,NumUnknownAuth,AmntUnknownAuth, NumUnknownReturn,AmntUnknownReturn*

3.7.2 Clear Failed History

Key Value: ADMIN
Admin Value: **CFH**
Description: allows user to easily perform house-cleaning functions against failed transaction log (GFT)
Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: standard
Response Table: 3.9 Administrative User Requests Response Table (pg 54)

3.7.3 Change Password

Key Value: ADMIN
Admin Value: **CHNGPWD**
Description: allows a user to change their own password
Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: standard
Response Table: 3.9 Administrative User Requests Response Table (pg 54)

3.7.4 Clear Transaction History

Key Value:	ADMIN
Admin Value:	**CTH**
Description:	allows user to easily perform house-cleaning functions against the settled transaction log (GL)
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.7.5 Close Batch

Key Value:	ADMIN
Admin Value:	**`closebatch`**
Description:	closes the currently open batch pending settlement (does not send to processor, still unsettled)
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.7.6 Automated Tasks (CRON)

Key Value:	ADMIN
Admin Value:	**`cron`**
Description:	allows user to schedule automated functions
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	comma delimited if **`cron=list`**, otherwise, standard response
Response Params:	*`cronid,cron_task,cron_date,last_ts,next_ts`*
Response Table:	3.9 Administrative User Requests Response Table (pg 54)
Reference:	A.3 Automated Task (CRON) Management

3.7.7 Edit Data Field

Key Value:	ADMIN
Admin Value:	**`fieldedit`**
Description:	For editing transaction details as stored in the Monetra database. This can edit fields that affect interchange (e.g tax, rate, bdate/edate, etc), or user-definable data (e.g. comments, clerkid, stationid). This process will only affect those transactions which are currently unsettled.
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Editable Fields	3.5 General User Request Parameters Table (pg 34)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.7.8 Force Batch Settle

Key Value:	ADMIN
Admin Value:	**`forcesettle`**
Description:	allows user to easily perform forced settlement of any open batch
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.7.9 Force Void

Key Value:	ADMIN
Admin Value:	**`forcevoid`**
Description:	allows a user to forcibly void a transaction from the unsettled batch when a standard void does not work. This should be used with caution.

Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: standard
Response Table: 3.9 Administrative User Requests Response Table (pg 54)

3.7.10 Get Permissions

Key Value: ADMIN
Admin Value: **`getperms`**
Description: gets permission for functionality allowed by this user/subuser
Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: comma delimited
Response Params: *`trantypes,admintypes,obscure,unattended`*

3.7.11 Failed Transactions

Key Value: ADMIN
Admin Value: **GFT**
Description: returns failed transactions
Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: comma delimited
Response Params: *ttid,msoft_code,phard_code,type,proc,entrymode,tranflags,card,abaroute,account,expdate,checknum,timestamp,code,verbiage,amount,cardholdername,avs,cv,clerkid,stationid,comments,divisionnum,promoid,descmerch,ptrannum,ordernum,custref,bdate,edate,roomnum,excharges,rate,raw_code,raw_avs,raw_cv,raw_cardlevel*

3.7.12 Settled Transactions

Key Value: ADMIN
Admin Value: **GL**
Description: returns all settled transactions for the requested user (merchant) account
Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: comma delimited
Response Params: *ttid,msoft_code,phard_code,type,proc,entrymode,tranflags,reversible,card,pclevel,cardlevel,abaroute,account,expdate,checknum,timestamp,amount,examount,tax,cashback,authnum,stan,batnum,item,cardholdername,avs,cv,clerkid,stationid,comments,divisionnum,promoid,descmerch,ptrannum,ordernum,custref,bdate,edate,roomnum,excharges,rate,raw_code,raw_avs,raw_cv,raw_cardlevel*

3.7.13 Unsettled Transactions

Key Value: ADMIN
Admin Value: **GUT**
Description: returns all unsettled transactions for the requested user (merchant) account
Access Level: Standard User
Param Table: 3.8 Administrative User Requests Parameters Table (pg 51)
Response Type: comma delimited
Response Params: *ttid,msoft_code,phard_code,type,proc,entrymode,tranflags,capture,card,pclevel,cardlevel,abaroute,account,expdate,checknum,timestamp,startdate,issuenum,amount,examount,tax,cashback,authnum,stan,batch,item,cardholdername,avs,cv,clerkid,stationid,comments,divisionnum,promoid,descmerch,ptrannum,ordernum,custref,bdate,edate,roomnum,excharges,rate,raw_code,raw_avs,raw_cv,raw_cardlevel*

3.7.14 Merchant Account Information

Key Value: ADMIN
Admin Value: **merchinfo**

Description:	Returns a row per cardtype with the name of the authorization entity, settlement entity, and associated merchant ids (for receipt printing) along with the transaction types that entity supports.
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	comma delimited
Response Params:	*cardtype,auth_proc,settle_proc,auth_merchid,settle_merchid, auth_sub,settle_sub,indcode,trantypes*

3.7.15 Post-Settlement Batch Totals

Key Value:	ADMIN
Admin Value:	**PBT**
Description:	returns batch totals for each settled batch in given date range
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	comma delimited
Response Params:	*BatchNum,sub,totaltransNum,totaltransAmount,totalAuthNum, totalAuthAmount,ReturnNum,totalReturnAmount,NumVisaAuth, AmntVisaAuth,NumVisaReturn,AmntVisaReturn,NumMCAuth, AmntMCAuth,NumMCReturn,AmntMCReturn,NumDiscAuth, AmntDiscAuth,NumDiscReturn,AmntDiscReturn,NumAmexAuth, AmntAmexAuth,NumAmexReturn,AmntAmexReturn,NumDinersAuth, AmntDinersAuth,NumDinersReturn,AmntDinersReturn,NumCBAuth, AmntCBAuth,NumCBReturn,AmntCBReturn,NumJCBAuth,AmntJCBAuth, NumJCBReturn,AmntJCBReturn,NumGIFTAuth,AmntGIFTAuth, NumGIFTReturn,AmntGIFTReturn,NumOtherAuth,AmntOtherAuth, NumOtherReturn,AmntOtherReturn,NumDebitAuth,AmntDebitAuth, NumDebitReturn,AmntDebitReturn,NumEBTAuth,AmntEBTAuth, NumEBTReturn,AmntEBTReturn,NumUnknownAuth,AmntUnknownAuth, NumUnknownReturn,AmntUnknownReturn*

3.7.16 Queue Checking

Key Value:	ADMIN
Admin Value:	**QC**
Description:	returns if a transaction(s) is still in the queue for user account
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	comma delimited
Response Params:	*queue*, *ordernum*, *clerkid*, *stationid*, *comments*, *ptrannum*

3.7.17 Renumber Batch

Key Value:	ADMIN
Admin Value:	**renumberbatch**
Description:	allows user to easily change current batch number, as requested, prior to settlement on those processors that Monetra maintains
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.7.18 Secure Transactions

Key Value:	ADMIN
Admin Value:	**securetrans**
Description:	secures the transactions in history so that they MAY NOT BE UNSETTLED
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.7.19 Set Batch Number

Key Value:	ADMIN
Admin Value:	**setbatchnum**
Description:	allows user to easily set the batch number used on those processors that Monetra maintains
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.7.20 Add Merchant Subuser

Key Value:	ADMIN
Admin Value:	**subuseradd**
Description:	allows merchant admin to add a subuser for processing tasks
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)
Reference:	A.1 User Administration

3.7.21 Delete Merchant Subuser

Key Value:	ADMIN
Admin Value:	**subuserdel**
Description:	allows the merchant admin to remove a subuser
Access Level:	Administrator
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)
Reference:	A.1 User Administration

3.7.22 Edit Merchant Subuser

Key Value:	ADMIN
Admin Value:	**subuseredit**
Description:	allows the merchant admin to add a subuser for processing tasks
Access Level:	Administrator
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)
Reference:	A.1 User Administration

3.7.23 List Merchant Subusers

Key Value:	ADMIN
Admin Value:	**`subuserlist`**
Description:	allows the merchant admin to add a subuser for processing tasks
Access Level:	Administrator
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	comma delimited
Response Params:	*`user,pwd,master,trantypes,admintypes,obscure,unattended`*
Reference:	A.1 User Administration

3.7.24 Unsettle Batch

Key Value:	ADMIN
Admin Value:	**`unsettlebatch`**
Description:	allows users to move transactions from the history DB to the Unsettled DB and re-submit batch to processor for approval
Access Level:	Standard User
Param Table:	3.8 Administrative User Requests Parameters Table (pg 51)
Response Type:	standard
Response Table:	3.9 Administrative User Requests Response Table (pg 54)

3.8 Administrative User Requests Parameters Table

Key	Req	Tran Type	Ref	Description
username	Y	all		General user (or subuser of a general user) username. Must NOT be 'MADMIN'
password	Y	all		password associated with user/subuser
action	Y	all		Always '**admin**'. User-administrative level requirement.
admin	Y	all		any Admin Value from preceding section
admintypes	C	subuseradd, subuseredit	A.1 User Administration	Administrative types to assign to a subuser. Separated by pluses or pipes (+ or |). Valid admintypes are any of the Admin Values from the preceding section.
acct	Opt	GUT, GL, GFT		Account number for auditing. This is not capable of partial number searches, because of the database encryption. Specify the whole card number. Please note that if searching the GFT report, or the searching a 'secured' batch in the GL report, you must pass the card number truncated as stored in the Monetra database. By default, if the card number was 5454545454545454, you must pass XXXXXXXXXXXX5454 as your auditing parameter to get results.
amount	C	fieldedit,GL,GFT,GUT		Amount to insert/update transaction on fieldedit, or amount to search for on reports.
batch	Opt	GUT, GL, CTH, BT, PBT, forcesettle, setbatchnum, renumberbatch, unsettlebatch		batch number to audit for. 'all' is NOT valid.
bdate	Opt	GUT, GL, CFT, CFH, CTH, fieldedit, securetrans, BT, PBT	A.13 Date Formats	begin date. Required if edate is specified.
capture	Opt	GUT		defaults to showing both captured and uncaptured transactions. specify as 'yes' or 'no' to show only captured or uncaptured transactions, respectively.
cardholder	Opt	GUT,GL,GFT		audit for user-defined field of

Key	Req	Tran Type	Ref	Description
name				'cardholdername'. This will perform a partial-match search.
cardtypes	Opt	GUT, GL, GFT	A.10 Card Types	Cardtypes to be returned in audit. Separated by pluses or pipes (+ or |).
clerkid	Opt	GUT, GL, GFT, QC, fieldedit		audit for user-defined field of 'clerkid'. This will perform a partial-match search. Or on fieldedit, will update clerkid field.
comments	Opt	GUT, GL, GFT, QC, fieldedit		audit for user-defined field of 'comments'. This will perform a partial-match search. Or on fieldedit, will update comments field.
cron	Y	cron	A.3 Automated Task (CRON) Management	Specify the cron function to perform. '**add**', '**list**', or '**remove**' are valid options.
cron_date	C	cron	A.3 Automated Task (CRON) Management	Date for cron formatted as described in appendix
cronid	C	cron	A.3 Automated Task (CRON) Management	Unique cron identifier as returned from audit
cron_task	C	cron	A.3 Automated Task (CRON) Management	the cron tasks available to User Administration are limited to '**SETTLE**', and custom functions. Please review the cron_task reference.
edate	Opt	GUT, GL, GFT, CFH, CTH, BT, PBT, fieldedit, securetrans	A.13 Date Formats	end date. Required if bdate is specified.
examount	Opt	fieldedit		Update/insert examount of transaction
excharges	Opt	fieldedit		Update/insert excharges of transaction
obscure	Opt	subuseradd, subuseredit		Yes/No value. Defaults to 'no'. If enabled, it will obscure account numbers from reports.
pclevel	Opt	GUT,GL		Limit reports by PCLevel, may pass 0, 1, or 2.
ptrannum	Opt	GUT, GL, GFT QC, fieldedit		Search for ptrannum in audit. Used to identify transaction to update for fieldedit.
pwd	C	chngpwd, subuseradd, subuseredit		Required on subuseradd. This is the password assigned to the sub account.
rate	Opt	fieldedit		Room rate for lodging. May be required

Key	Req	Tran Type	Ref	Description
				for settlement on some transactions.
reversible	Opt	GL		whether the transaction in the batch may be 'unsettled' or 'unvoided'. Takes boolean values with special value of 'both'. (Defaults to both)
showvoids	Opt	GL		whether the report should include voids. Boolean with special value of 'only' to show only voids. (Defaults to yes)
stationid	Opt	GUT, GL, GFT, QC, fieldedit		audit for user-defined field of 'stationid'. This will perform a partial-match search. Or on fieldedit, will update stationid field.
sub	C	forcesettle, renumberbatch,set batchnum		Assuming split-route account, the sub account number to affect
tax	Opt	fieldedit		Insert/Update tax field
trantypes	Y	subuseradd,subuseredit		list of transaction types you wish to allow sub user to utilize. This list should be delimited by pipes or pluses ('|' or '+')
ttid	Opt	GUT, GL, GFT, fieldedit, forcevoid		For audits, will only return the specific transaction specified by ttid. For fieldedit, will edit the transaction specified. For forcevoid will forcibly remove the transaction from Monetra.
type	Opt	GUT, GL, GFT		Only return list of transactions that match the type specified by this field. May specify multiple types separated by | or +.
unattended	Opt	subuseradd, subuseredit		If account is added as unattended, it will not be subject to password expiration
user	Y	subuseradd, subuseredit, subuserdel		Sub-user username to manage

3.9 Administrative User Requests Response Table

Key	*Tran Type*	*Ref*	*description*
code	ALL	A.4 Result Codes (code)	result code of transaction
msoft_code	ALL	A.5 Result Codes (msoft_code)	detailed result code specific to Monetra-internal checks
verbiage	ALL		textual, human-interpretable response code (meant for system display to clerk)

A Appendices

A.1 User Administration

A.1.1 Monetra User Subsystem Overview

Monetra is designed with a fully virtual database-backed user system. Upon initial installation, a single user account is present known as *MADMIN*, which is the Monetra engine administrator. The default password for this user is '*password*'. Every user except *MADMIN* is tied to at least one merchant account for processing. All users may have an unlimited number of *subusers*. A subuser can affect the same data as the *master* user that created it, but it may also be limited to allow only a certain subset of transactions, and may also be set up to obscure privileged information such as account numbers, or may be set as attended or unattended (which only affects password expirations).

Once user accounts are added, they may be edited, deleted, enabled or disabled. Disabling a user account simply prevents the ability of that user to log into the system and run transactions.

A.1.2 Information Required for Adding Users

There are a few data fields available when adding a user account, independent of the selected processor.

- **`user`** Username to assign to account (Must be specified, no default)
- **`pwd`** Password to assign to user (Must be specified, no default)
- **`profile_id`** Profile identification as stored on Main Street's servers to identify which merchant account this user belongs.
- **`proc`** The name of the processing institution to which the account is bound. (Must be specified, no default)
- **`indcode`** One of the supported industry codes (Must be specified, no default)
- **`cardtypes`** specify cardtypes to accept for this accounts (Defaults to **`ALLCREDIT`**)
- **`mode`** specify whether this account does authorizations, settlements or both (Defaults to **`BOTH`**)
- **`email`** specify an e-mail address where cron-tasks will send information (Defaults to blank)
- **`conn_priority`** specify an *override* for the default conn_priority configured for the engine. This should usually be left blank, except under special circumstances where you have multiple merchant accounts talking to the same processing institution, but they cannot utilize the same connectivity method (e.g. First Data requires a Datawire ID to utilize their Internet gateway. If you were not assigned a Datawire ID on one of the accounts, it may be necessary to only utilize the modem for that account).

When adding users to Monetra, you will also be providing merchant setup information. It is recommended that you pull a **`proclist`** transaction for a list of available processors (with capabilities), then issue a **`procfields`** transaction to return the required data elements with formatting information that must be passed with the transaction. An alternative to this (especially for releases of Monetra prior to 5.1), is to reference the available parameters from the Monetra User Setup Reference available at http://www.monetra.com/documentation.html . Monetra will deny the request if all required fields are not presented and formatted correctly.

A.1.3 Binding Multiple Merchant Accounts to One User

In some circumstances, it is desirable to bind multiple merchant accounts to a single user account. Specific examples include needing to authorize through one processing institution, but settling through another. Another example might be if you want to process credit cards through one merchant account, but gift cards through another. Monetra can internally handle the routing of these transactions without having to set up separate user accounts. The base sub account is always 0. If no **sub** identifier is sent, 0 is assumed. If the 0 sub account is deleted, the entire user account will be deleted, including any transaction history associated with it. The number chosen for the **sub** field does not have to be synchronous; as long as 0 exists, you may use any other positive values you wish.

Monetra refers to this type of merchant account binding as adding *sub accounts*. Sub accounts must differ by the supplied '**mode**' and/or '**cardtypes**' and '**sub**' fields. You may not have two sub accounts with the same cardtypes and same mode. If a sub account differs from other accounts by the mode, the 'ALL' or 'BOTH' modes must not be utilized.

An example of a complex sub account would be:

```
sub=0, mode=AUTH, cardtypes=ALLCREDIT, proc=VITAL [,...]
sub=1, mode=SETTLE, cardtypes=ALLCREDIT, proc=SALEM [,...]
sub=2, mode=BOTH, cardtypes=ALLGIFT, proc=SVS [,...]
```

Notice that the sub 2 account differs by the cardtypes, so a mode of BOTH is valid here.

A.1.4 Sub Users

Sub users are used if you want to assign different permission levels to different people that have access to your merchant account. This is a recommended procedure if more than one person will have access to the Monetra system, as you should only grant individuals the access levels they require to perform their duties. Note that there is a major difference between sub users and sub accounts, do not confuse the two. Sub users deal with permissions, sub accounts deal with processing.

When logging in as a sub user, the username must be prefixed with the username of the user that added it, followed by a colon. We'll call this the master user. For instance, if the username was '`merchacct1`', and you had a sub user named '`bob`', the login name would be '`merchacct1:bob`'. Though when adding or editing the account as the master user, you may reference the user as simply '`bob`' in the user field, since Monetra can automatically determine the full name (though it may also be referenced by the full name). Available parameters.

- **`user`** Sub user name to add to 'master' user
- **`pwd`** Password to assign sub user
- **`trantypes`** These are the action keys allowed separated by either pipes (|) or pluses (+) for this user account. Depending on if you're adding an MADMIN sub user or a sub user for a merchant account, you may need to reference different sections of this guide.
- **`admintypes`** Specify the administrative functions this sub user is allowed to access, this is only relevant for non-MADMIN accounts, (e.g. standard merchant accounts), and references the admin keys in 3.8 Administrative User Requests Parameters Table
- **`obscure`** Boolean (yes or no) value on whether or not sensitive data (account numbers) should be obscured in the reports.
- **`unattended`** Unattended accounts are not subject to password expiration

A.2 Login Restriction Management

A.2.1 Restriction Overview

Monetra supports restricting logins by validating SSL client certificates on a per-user basis. Monetra must already be configured to accept and validate client certificates at the engine level. This section assumes you already manage your own CA (Certificate Authority), and have general knowledge about SSL and certificate management.

A.2.2 Adding, Removing, and Listing restrictions

The administrator can add per-user restrictions on which client certificate(s) are allowed to execute transactions. Monetra uses a cryptographically-secure digest (hash) of the client certificate to identify individual client certificates.

The client certificate digest can be generated using the *M_SSLCert_gen_hash()* as provided by LibMonetra, giving the filename as the first argument. Alternatively, the digest can be generated using the *X509_digest()* digest in OpenSSL with the SHA1 digest algorithm or can be generated using the openssl(1) application:

```
$ openssl x509 -sha1 -in newcert.pem -noout -fingerprint
fingerprint: 96:f8:ac:6b:76:8b:d5:f3:5f:bb:2d:0c:4e:9d:19:c4:b4:49:ad:36
$
```

This fingerprint is then passed along to the **restriction_data** parameter. The only currently supported **restriction_type** is **ssl_cert .** All user accounts in Monetra may be assigned ssl certificate restrictions.

Refer to the respective parameters table for add, list, and remove options.

A.3 Automated Task (CRON) Management

A.3.1 Cron Overview

Monetra has a built-in task scheduler known by the common Unix name of 'cron'. It is capable of scheduling various tasks such as settlements, database vacuum/management, and even generic/user-definable functions on a periodic basis. There are 2 interfaces to the CRON subsystem. The first is the Engine Administrator level, where the most common tasks are database maintenance and automated sending of user statistics. The second is at the user-level, which is mainly used for automating settlements.

The cron subsystem will e-mail the results of each task to the specified locations. For User Administrative tasks, it will use the e-mail addresses configured in the merchant setup. For Engine Administrative tasks, it will use the email addresses specified in the main Monetra configuration. Monetra must be properly configured to send e-mails either via a local instance of *sendmail* or a valid SMTP server location.

A.3.2 Date Formatting

The date formatting is very flexible to allow a wide range of automated task scheduling. You can issue tasks that run multiple times daily, or once a month. The format is:

<time>[;<time>[;...]]|<day/date>[;<day/date>[;...]]

<time> is represented as HHMM, assuming a 24hr clock. (e.g 1430 is 2:30pm)
<day/date> is represented as standard three letter abbreviations for the day of the week, or are represented as the numeric day of the month starting at 1
An asterisk (*****) represents every day, which is easier than specifying each day of the week.

Example Formats

Format	*Meaning*
`0100|fri`	1:00 am every Friday
`0000;1200|mon;thu`	12:00am and 12:00pm every Monday and Thursday
`1200|1;15`	12:00pm on the 1st and 15th of each month
`2200|*`	10:00pm everyday

A.3.3 Cron Tasks

Engine administrators may specify '**DBMAINT**' as the **cron_task** to perform database maintenance such as vacuums on a regular basis, or may specify a custom cron task.

User administrators may specify '**SETTLE**' as the **cron_task** to perform an automated settlement for the user account (for all open batches, which will be settled one at a time).

A custom **cron_task** must be formatted as a full transaction as outlined in the Monetra IP, SSL, and DropFile Specification, but any new-line characters must be represented as the characters '**\n**'. The username and password will be automatically pre-pended to the transaction, so should NOT be specified. Beware that the cron tasks are not stored encrypted, so you MUST NOT embed sensitive data such as account numbers!

A.4 Result Codes (code)

Result Code	*Description*
AUTH	transaction authorized
CALL	call processor for authorization
DENY	transaction denied
DUPL	duplicate transaction
PKUP	confiscate card
RETRY	retry transaction
SETUP	setup error
TIMEOUT	transaction not processed in allocated amount of time

A.5 Result Codes (msoft_code)

Result Code	*Description*
INT_SUCCESS	All Monetra test passed
UNKNOWN	Unknown/ unset. Could be pass or fail
INT_GENERICFAIL	Generic/ undefined failure
ACCT_DISABLED	Account disabled
ACCT_INVALIDTRANS	Invalid transaction type for user
ACCT_PASS	Password invalid
ACCT_PASSEXPIRED	Password reached expiration
ACCT_SSLCERT	SSL Certification check failed
ACCT_TOOMANYATTEMPTS	Too many bad login attempts
ACCT_TRANSNOTALLOWED	User does not have permission for transaction type
ACCT_USER	Username not found
DB_FAIL	Failure to write to Monetra database
CONN_MAXATTEMPTS	Maximum attempts to connect to processor reached
CONN_MAXSENDS	Maximum send attempts reached
CONN_TOREVERSAL	TOReversal must be issued. Status of transaction received unknown
DATA_ACCOUNT	Bad account number
DATA_AMOUNT	Bad amount
DATA_BATCHLOCKED	Batch locked
DATA_EXPDATE	Bad expiration date
DATA_INVALIDMOD	Invalid modification to existing transaction
DATA_NOOPENBATCHES	No open batches/ batch not found
DATA_RECORDNOTFOUND	Record not found
DATA_TRACKDATA	Bad Trackdata
DB_FAIL	Critical Database Failure
LIC_CARDTYPE	License does not allow that cardtype
LIC_TRANEXCEED	License Transaction Limit has been exceeded
LIC_USERS	Max user accounts reached
SETUP_CARDTYPE	Cardtype not in setup
SETUP_DATA	Generic setup issue
SETUP_SCHED	Transaction could not be scheduled
SETUP_TRANTYPE	Trantype not supported for merchant
SNF	This transaction has been authorized by the store and forward system

Result Code	*Description*
SYS_MAINTENANCE	Transaction type not allowed in maintenance mode
SYS_SHUTDOWN	Shutdown being attempted

A.6 Result Codes (phard_code)

Result Code	*Description*
SUCCESS	Generic success
UNKNOWN	Unknown- could be pass or fail
GENERICFAIL	Generic/ undefined failure
ACCTERROR	Account number or length error
ACCOUNT_CLOSED	Account has been closed
ALREADY_ACTIVE	Account has already been activated
ALREADY_REVERSED	Reversal already issued
BAD_MERCH_ID	Bad merchant ID
BAD_PIN	Bad Debit/ EBT pin info.
BALANCE_MISMATCH	Settlement totals did not match host
CALL	Call issuer for authorization
CARD_EXPIRED	Credit card expired
CASHBACK_EXCEEDED	Too much cash back
CASHBACK_NOAVAIL	Cash back services unavailable
CID_ERROR	CVV2/ CID error
DATE_ERROR	Date error
DONOTHONOR	Do not honor card
DUPLICATE_BATCH	Duplicate batch number
ENCRYPTION_ERROR	Encryption Error (usually debit/ebt)
EXCEED_ACTIVITY_LIMIT	Exceeds activity limit
EXCEED_WITHDRAWAL_LIMIT	Exceeds withdrawal limit
ID_ERROR	Valid ID required for transaction
INSUFFICIENT_FUNDS	Insufficient funds
INVALID_SERVICE_CODE	Invalid service code
MANAGER_NEEDED	Manager needed (possibly velocity warning)
NOREPLY	No reply from processor back end
NOT_ACTIVE	Account has not yet been activated
NOT_PERMITTED_CARD	Card not permitted for this transaction type
NOT_PERMITTED_TRAN	Transaction type not permitted for this account
PICKUP_FRAUD	Confiscate card (fraud assumed)
PICKUP_LOST	Confiscate card (reported lost)
PICKUP_NOFRAUD	Confiscate card (no fraud assumed)
PICKUP_STOLEN	Confiscate card (reported stolen)

Result Code	*Description*
RECURRING_CANCEL	Cardholder requested a cancellation on recurring charges
REENTER	Bad transaction data or setup, reenter
REJECTED_BATCH	Batch rejected for settlement
REPRESENTED	Represented Check
SECURITY_VIOLATION	Security violation
SYSTEM_ERROR	Generic system error
VIOLATION	Violation

A.7 Result Codes (AVS)

Result Code	*Description*
BAD	All checks failed
GOOD	Good avs result
STREET	Street verification failed
UNKNOWN	Result unknown, should treat as good
ZIP	Zip code verification failed

A.8 Result Codes (CV)

Result Code	*Description*
BAD	Verification failed
GOOD	CV verification success
UNKNOWN	Result unknown- should treat as good

A.9 Visa Card Level Results

LEGEND: ^ = space

Monetra CardLevels	*Raw Code*
VISA_TRADITIONAL	A^
VISA_TRADITIONAL_REWARDS	B^
VISA_SIGNATURE	C^
VISA_INFINITE	D^
RESERVED	E^
RESERVED	F^
VISA_BUSINESS	G^
VISA_CHECK	H^
VISA_COMMERCE	I^
RESERVED	J^
VISA_CORPORATE	K^
RESERVED	L^
MASTERCARD_EUROCARD_DINERS	M^
RESERVED	N^
RESERVED	O^
RESERVED	P^
PRIVATE_LABEL	Q^
PROPRIETARY	R^
VISA_PURCHASE_CARD	S^
INTERLINK	T^
VISA_TRAVELMONEY	U^
RESERVED	W^
RESERVED	X^
RESERVED	Y^

Monetra CardLevels	*Raw Code*
RESERVED	Z^
RESERVED	0^
RESERVED	1^
RESERVED	2^
RESERVED	3^
RESERVED	4^
RESERVED	5^
RESERVED	6^
RESERVED	7^
RESERVED	8^
RESERVED	9^
VISA_SIGNATURE_BUSINESS	G1
VISA_BUSINESS_CHECK	G2
VISA_GENERAL_PREPAID	J1
VISA_PREPAID_GIFT	J2
VISA_PREPAID_HEALTH	J3
VISA_PREPAID_COMMERCIAL	J4
VISA_GSA_CORPORATE_TANDE	K1
PRIVATE_LABEL_PREPAID	Q1
VISA_PURCHASE_FLEET	S1
VISA_GSA_PURCHASE	S2
VISA_GSA_PURCHASE_FLEET	S3
RESERVED	V1
AMEX	AX
DISCOVER	DI

A.10 Card Types

Type	*Class*	*Description*
VISA	Credit	Visa
MC	Credit	MasterCard
AMEX	Credit	American Express
DISC	Credit	Discover
DINERS	Credit	Diners Club
CB	Credit	Carte Blanche
JCB	Credit	JCB
SWITCH	Credit	Switch/Solo
BML	Credit	Bill Me Later
CHECK	Check	Electronic Checks
GIFT	Gift	Generic Gift
OTHER	Gift	Generic Gift/Loyalty
VISADEBIT	Debit	Debit in VISA bin-range
VISADS	Debit -> Credit	Transaction originally ran as Debit, but processor reported back that it was actually run as a credit card (Visa Only)
MCDEBIT	Debit	Debit in Mastercard bin-range
OTHERDEBIT	Debit	Debit not in Visa or Mastercard bin-range
EBT	EBT	EBT (electronic benefits transfer -- US foodstamps/wellfare)
ALL	Credit,Debit,EBT,Gift	Aggregate macro for all card types
ALLCREDIT	Credit	Aggregate macro for all credit card types
ALLDEBIT	Debit	Aggregate macro for all debit card types
ALLGIFT	Gift	Aggregate macro for all gift card types
ALLEBT	EBT	Aggregate macro for all EBT card types (only one though)

A.11 Excharges Values

Result Code	*Description*
REST	Restaurant/Room Service charges
GIFT	Gift charges
MINI	Mini-fridge charges
TELE	Telephone Charges
LAUND	Laundry Charges
OTHER	Other Charges

A.12 Industry Codes

Code	*Description*
A	(future)
AE	(future)
AF	Automated Fueling
E	E-Commerce
F	Food/Restaurant Retail
FE	Food/Restaurant E-Commerce (ask for more information!)
G	Grocery (ask for more information!)
GE	Grocery E-Commerce (ask for more information!)
H	Hotel/Lodging
M	MailOrder/PhoneOrder
R	Retail
RS	Retail Self-Serve (Kiosk)

A.13 Date Formats

These date formats apply to the `bdate` and `edate` fields.

A.13.1 Special Keywords

Name	*Meaning*
now	current date/time
epoch	Unix timestamp of 0 = Jan 1, 1970 00:00:00 UTC

A.13.2 Offset format

Offsets take the format of:

+ or -
magnitude
space
modifier

Modifiers
year[s]
month[s]

Modifiers
week[s]
day[s]
hour[s]
min[s|ute|utes]
sec[s|ond|onds]

Example: “+1 day” or “-5 years”

A.13.3 Explicit date formats

Formats
YYYY-MM-DD
YYYY/MM/DD
YYYY-MM-DD hh:mm
YYYY-MM-DD hh-mm
YYYY/MM/DD hh:mm
YYYY/MM/DD hh-mm
YYYY-MM-DD hh:mm:ss
YYYY-MM-DD hh-mm-ss
YYYY/MM/DD hh:mm:ss
YYYY/MM/DD hh-mm-ss
MM-DD-YYYY
MM/DD/YYYY
MM-DD-YYYY hh:mm
MM-DD-YYYY hh-mm
MM/DD/YYYY hh:mm
MM/DD/YYYY hh-mm
MM-DD-YYYY hh:mm:ss
MM-DD-YYYY hh-mm-ss
MM/DD/YYYY hh:mm:ss
MM/DD/YYYY hh-mm-ss
MM-DD-YY
MM/DD/YY
MM-DD-YY hh:mm
MM-DD-YY hh-mm
MM/DD/YY hh:mm

Formats
MM/DD/YY hh-mm
MM-DD-YY hh:mm:ss
MM-DD-YY hh-mm-ss
MM/DD/YY hh:mm:ss
MM/DD/YY hh-mm-ss
MMDDYYYY
MMDDYY

A.14 Debug Levels

Name	*PCI Compliant?*	*Meaning*
INIT	Y	Basic initialization info, such as Monetra version
CONF	Y	Show configuration and startup details
WARN	Y	Warnings such as misconfiguration, etc.
INFO	Y	Uncategorized short information (like stats maybe)
TRAN	Y	Basic information on when a transaction enters the queue and is finished
TRAN_DETAIL	Y	Basic incoming parameters as parsed (with sensitive data removed), same for outgoing response parameters. Probably will imply INFO.
TRAN_TRACE	N	Parsed incoming transaction requests unobscured, same with outgoing responses
CONN	Y	Log connection details such as IP address, when it is opened, closed, etc.
PROC	Y	Log connection info to processors and details when they process a transaction
PROC_DETAIL	Y	Log more detailed, sanitized, information such as the sanitized traces.
PROC_TRACE	N	Non-sanitized version of PROC_DETAIL, just prints ASCII representation of protocol. Not as useful as TRACE_OUT for binary protocols.
TRACE_IN	N	Raw trace of data in and out from client connections
TRACE_OUT	N	Raw trace of data between Monetra and the processors
SQL	N	SQL Statements
ERROR	Y	Any significant error condition

Name	*PCI Compliant?*	*Meaning*
CRIT	Y	A critical/significant error which must not be ignored as it will severely affect the running system
DEBUG	N	Ungrouped Debug data, very verbose
DEV	N	Very verbose debug information in place for development of new features. These should not exist in a production release.

A.15 General User Request Examples

A.15.1 Example: Sale

Request Keys	Request Values
username	vitale
password	test123
action	sale
account	4012888888881
expdate	0512
amount	12.00
street	8320
zip	85284
cv	999
comments	test transaction
ptrannum	123456
Response Keys	**Response Values**
code	AUTH
msoft_code	INT_SUCCESS
phard_code	SUCCESS
auth	123456
avs	GOOD
cv	GOOD
item	1
batch	1
verbiage	APPROVED
timestamp	1121867675
ttid	112

A.15.2 Example: Void

Request Keys	Request Values
username	vitale
password	test123
action	void
ttid	112
Response Keys	Response Values
code	AUTH
msoft_code	INT_SUCCESS
ttid	112
verbiage	SUCCESS

A.16 Engine Administrative Request Examples

A.16.1 Example: Add User

Request Keys	Request Values
username	MADMIN
password	password
action	adduser
user	vitale
pwd	test123
proc	vital
indcode	E
cardtypes	VISA|MC|AMEX
mode	BOTH
email	support@monetra.com
merchid	123444444123
bankid	111119
vnumber	00000001
storeid	0001
termid	0001
agentid	111111
chainid	000000
merchcat	5999
merchname	Main Street
merchloc	Gainesville
statecode	FL
servicephone	321-2517794
Response Keys	Response Values
code	AUTH
msoft_code	INT_SUCCESS
verbiage	USER ADDED

A.16.2 Example: List Users

Response Keys	Response Values
username	MADMIN
password	password
action	listusers
Response Data	
userlist,user status,master,trantypes,admintypes vitale,E,1,, vitale:test,E,0,SALE|PREAUTH|VOID|ADMIN,GUT	

A.17 Administrative User Request Examples

A.17.1 Example: Get Settled Transactions

Request Keys	Request Values
username	vital
password	test123
action	admin
admin	GUT
bdate	01-01-05
edate	01-02-05
Response Data	
ttid,msoft_code,phard_code,type,capture,pclevel,card,account,expdate,startdate,issuenum,amount,examount,tax,cashback,authnum,stan,timestamp,batch,item,cardholdername,avs,cv,clerkid,stationid,comments,divisionnum,ptrannum,raw_code,raw_avs,raw_cv 112,INT_SUCCESS,SUCCESS,SALE,yes,0,VISA,4012888888881,0512,,,12.00,0.00,0.00,0.00,123456,,01/02/05 12:16,1,1,,GOOD,GOOD,,,test transaction,,12345,00,Y,M	

A.17.2 Example: Renumber Batch

Request Keys	Request Values
username	vital
password	test123
action	admin
admin	renumberbatch
batch	1
newbatch	3
Response Keys	**Response Values**
code	AUTH
msoft_code	INT_SUCCESS
verbiage	BATCH RENUMBERED