Monetra®

Application Interface Guide

Revision: 8.17.0
Publication date June 29, 2021

Application Interface Guide

Monetra Technologies, LLC

Revision: 8.17.0

Publication date June 29, 2021

Legal Notice

The information contained herein is provided *As Is* without warranty of any kind, express or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. There is no warranty that the information or the use thereof does not infringe a patent, trademark, copyright, or trade secret.

Monetra Technologies, LLC. SHALL NOT BE LIABLE FOR ANY DIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, WHETHER RESULTING FROM BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, OR OTHERWISE, EVEN IF MONETRA TECHNOLOGIES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. MONETRA TECHNOLOGIES RESERVES THE RIGHT TO MAKE CHANGES TO THE INFORMATION CONTAINED HEREIN AT ANYTIME WITHOUT NOTICE. NO PART OF THIS DOCUMENT MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, FOR ANY PURPOSE, WITHOUT THE EXPRESS WRITTEN PERMISSION OF Monetra Technologies, LLC.

Table of Contents

1 Revision History

Version	Date	Changes
8.17.0	2021-06-29	• Corrected `amount` description and format for recurring payments • Updated `nsf` request parameter to indicate that partial authorizations are automatically reversed when merchant does not explicitly allow them • Added Monetra Code `NSFAUTODENY` • Added customer flag `email_receipt_recurring` • Added BIN range format appendix section • Updated general formatting
v8.17.0	2021-04-20	• Changed revision history ordering to most recent first • Detailed fields received in datablocks for all actions in Chapter 4: *Admin User Actions* • Specified that `merch_custom_fields`, `merch_customer_fields`, and `custom_customer_fields_spec` field names are lowercase • Updated push notification information in Section 4.12: Push Notification Management
v8.16.0	2021-04-15	• Updated formatting • Detailed fields received in datablocks for all actions in Chapter 5: *Main Merchant User Actions* and Chapter 6: *Merchant Subuser Actions* • Added `descloc` and `cavvresp` response fields to Section 5.6.2: Get Unsettled Transactions, Section 5.6.3: Get Settled Transactions, and Section 5.6.4: Get Failed Transactions reports • Added note on how to properly parse datablocks
v8.13.0	2019-12-06	• Complete rewrite
v8.0.0	2017-06-29	• Initial document re-write. Now includes information from legacy addendums: IP/SSL/Drop File, XML spec, DSS Storage and billing

2 Overview

2.1 Introduction

Monetra is a fast, efficient, and secure payment application that is certified [https://www.monetra.com/certifications] to connect many types of applications directly to any of the major North American-based payment processors. It's designed to scale from small, custom, embedded devices to fully-redundant payment servers processing thousands of transactions per minute. Trusted for over 15 years by thousands of merchants throughout North America, Monetra is the premier product of its type.

Monetra supports critical payment features such as:

- Extensive EMV Processor certifications across the US and Canada
- Robust Tokenized Card Storage and Recurring Billing
- P2PE Card Encryption proven to comply with PCI P2PE standards, including HSM support
- Flexible, developer-friendly integration options, including ReSTful APIs and iFrame integrations
- Clustering support for redundancy and load balancing

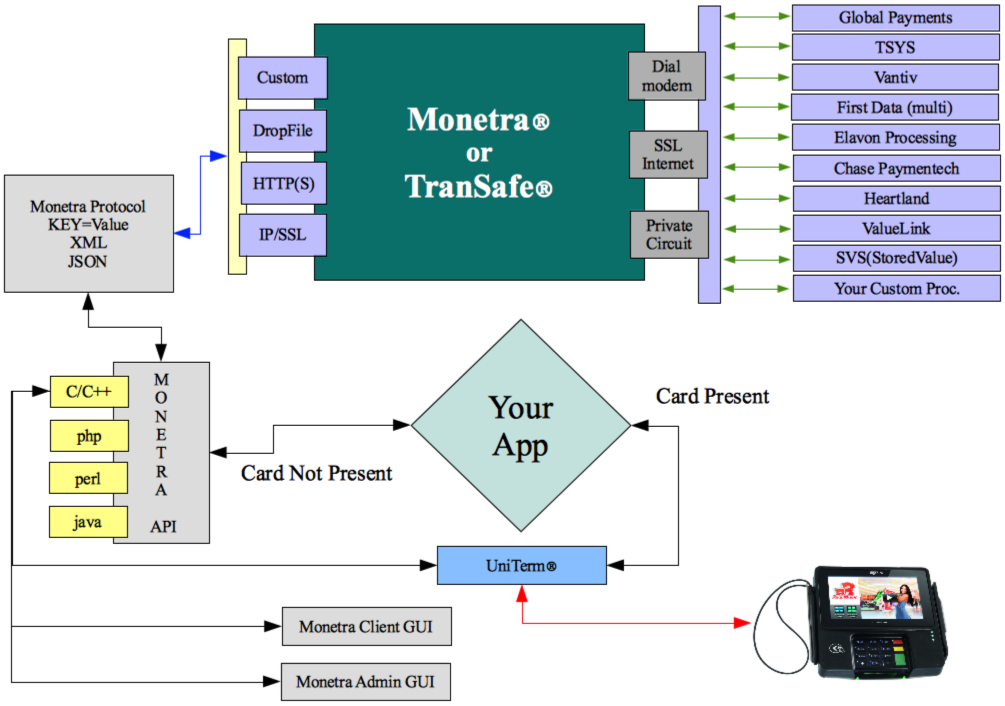
Designed from the ground up as a true Client/Server application, Monetra is written in 100% ANSI C89 to be compatible across all major and embedded operating systems, making it one of the most efficient, scalable, and portable payment engines available. It is built on a small payment core that houses all security and routing details, with all other features implemented via our flexible modular subsystem, which acts as an abstraction layer so that a change to the core does not impact a module and vice-versa. Inbound communications protocols, databases, processing institutions, etc. are all designed as separate modules to a common interface.

By providing a mature and feature-rich protocol, Monetra enables developers to build advanced payment-acceptance support directly into their application, including desktop, web, and mobile applications. Integrations can connect to Monetra using XML or JSON, both over HTTPS, or through the ReST API.

Monetra has been under constant development for over eighteen years while incorporating the functionality of our clients' best input directly into the production product along the way. We work hard to add more and more features and functionality to the core payment server every day. If there are any features you would like to see included, please feel free to contact sales@monetra.com.

2.2 Architecture

Figure 2.1.



Architecture: Monetra and TranSafe

2.3 Users

The user subsystem is split into two separate groups: Admin Users, who administer Monetra; and Merchant Users, who use the system to run payments. Within both of these groups, users are divided by level of permission granted; the main user will have elevated permissions, while the subusers will have restricted permissions.

Note: For security-related segmentation purposes, each user group will send transactions to Monetra via a different port: Admin Users will send requests to port 8666, while Merchant Users will send requests to port 8665, as per our IANA assignments.

2.3.1 Admin Users

When Monetra initializes its database for the first time, it creates a default user of `MADMIN` with a password of `password`. This default password is marked as expired when created and can only be used to change the password; all other operations are prohibited until the password is changed. This account will be used to create the initial Admin Users and then never be used again. You should create an Admin User for each person to whom you wish to grant that level of access. It is required that at least one administrator be granted ALL privileges for Administrator and Auth-As functionality (which creates a superuser), since the initial MADMIN user will not be used after this point.

An Admin User can perform a number of actions, including:

- Create Merchant Users
- Enable, disable, and configure Processors
- Get information about the system and run reports
- Schedule cron tasks
- Import/export the database

For a complete list of actions and detailed instructions on each, see Chapter 4: *Admin User Actions*.

Note: Because all Admin Users are technically a subuser of MADMIN, they will be referenced in the `username` field for Admin requests as `madmin:username`, where `username` is the Admin User in question.

2.3.2 Merchant Users

When a Merchant User is initially created, it will have all permissions enabled. Once the account is set up (see the Secure Install Guide for instructions on that) and a main user with all permissions has been created, the account password will not be used again and should be destroyed. If needed in the future, it can be reset by an Admin User. The main user can then create subusers. In general, the only difference between a main Merchant User and a Merchant Subuser lies in the permissions. For example, main users have the appropriate permissions to run **`action`**=**`admin`** requests as outlined in Chapter 5: *Main Merchant User Actions*, while subusers typically only run the transaction requests in Chapter 6: *Merchant Subuser Actions*.

The main Merchant User can perform a number of administrative actions, including:

- Manage Merchant Subusers
- Manage batches
- Run reports

For a complete list of main Merchant User actions and detailed instructions on each, see Chapter 5: *Main Merchant User Actions*.

The standard Merchant Subuser can mostly perform transactions, including:

- Run Credit/Debit Card Sales
- Handle Gift Cards
- Convert Checks

For a complete list of standard Merchant Subuser actions and detailed instructions on each, see Chapter 6: *Merchant Subuser Actions*.

Note: You can add as many Merchant Subusers as needed.

2.3.3 Permissions

A user/subuser can perform only the actions for which they have explicit permissions. For example, If an Admin User/Subuser needs to create a new Merchant User, then they need to have the `ADDUSER` permission in `madmintypes`. Similarly, if a Merchant User/Subuser wants to run a Return, then they need the permission `RETURN` in `trantypes`. Permission sets are merely the actions that a user or subuser is allowed to run. To run an action, the user/subuser must have that permission in their permission set.

2.4 System Authentication

Each user will authenticate with a `username` and `password`. These passwords should be set up in accordance with current PCI guidelines for strength and past-usage requirements. In addition to these credentials, an SSL certificate may be used to enhance security. For more information on setting up SSL certificates, see the Secure Install Guide.

2.4.1 Admin User Authentication

To run Administrative actions, Admin Users will authenticate with a `username` and `password`. These passwords should be set up in accordance with current PCI guidelines for strength and past-usage requirements.

Request Transaction		
Key	**Req**	**Description**
`username`	Y	A main Admin User or an Admin Subuser
`password`	Y	Password associated with user/subuser

Example: In the example below, we will be using the Admin User `fred` with a password of `P@ssw0rd123`.

```
username = madmin:fred
password = P@ssw0rd123
```

Note: Because all Admin Users are technically a subuser of MADMIN, they will be referenced in the `username` field for Admin requests as `madmin:username`, where `username` is the Admin User in question.

2.4.1.1 Auth-As Authentication

In addition to running Administrative actions, Admin Users with the appropriate permissions can run transactions as a Merchant User using the "auth-as" subsystem. This allows administrators to use the system as a particular Merchant User without needing to know the user's password.

Request Transaction		
Key	**Req**	**Description**
`username`	Y	A main Admin User or an Admin Subuser
`password`	Y	Password associated with user/subuser
`auth-as`	Y	Merchant User to authenticate as

Example: In the example below, we will be using the Admin User `fred` with a password of `P@ssw0rd123` to use the `Store5` Merchant account as Merchant Subuser `clerk1`.

```
username = madmin:fred
```

```
password = P@ssw0rd123
auth-as = Store5:clerk1
```

The shorthand for authenticating as a Merchant User is as follows:

```
Admin_User[:Admin_Subuser]::Merchant_User[:Merchant_Subuser]
```

Continuing with the previous example, the shorthand notation would be used like this:

```
username = madmin:janet::Fredstractors:clerk1
password = P@ssw0rd123
```

Note: Auth-As can be handy when you have hundreds or thousands of merchants on your system and need to perform a certain action for a Merchant account (such as settle a batch or pull a failed report for research).

2.4.2 Merchant User Authentication

All Merchant Users will authenticate with a username and password. These passwords should be set up in accordance with current PCI guidelines for strength and past-usage requirements.

Request Transaction		
Key	**Req**	**Description**
username	Y	A main Merchant User or a Merchant Subuser
password	Y	Password associated with user/subuser

Example: In the example below, we will be using the Merchant account Bobsboats as Merchant Subuser jimmy with a password of P@ssw0rd123.

```
username = Bobsboats:jimmy
password = P@ssw0rd123
```

3 Monetra Integration

3.1 Definitions

Identifier - A unique string of numbers and/or letters assigned to the transaction. This identifier may be repeated once a response has been given for the original transaction using the identifier.

Message - This is the actual data of a transaction that Monetra interprets and processes. Any data outside of this is strictly for transport/communication.

Start of Transaction - Indicator representing the start of a transaction, commonly known as `STX`. Hex value 0x02, decimal value: 2.

End of Transaction - Indicator representing the end of a transaction, commonly known as `ETX`. Hex value 0x03, decimal value: 3.

Field Separator - Character to separate one major portion of a message from another, commonly known as `FS`. Hex value 0x1C, decimal value: 28.

3.2 Communication and Transport

Monetra boasts a true Server/Client architecture, which allows for all system functionality to transpire either on the local machine or remotely across many dispersed systems. The connection methods that Monetra uses are completely modular by design, allowing for custom integration modules to be created and added easily. Modules are loaded at runtime. Currently, integrations can connect to Monetra over TLS/SSL only and format the message using either HTTP-POST with an XML or JSON payload, the ReST API, or LibMonetra.

3.2.1 TLS/SSL

A standard TLS/SSL connection should be made to the Monetra server. Transactions may be sent immediately upon successful connection.

For security-related segmentation purposes, each user group sends transactions to Monetra via a different port. These are the default port numbers for communication with Monetra:

Port	User Group	Actions
`8666`	Admin Users/Subusers	Chapter 4: *Admin User Actions*
`8665`	Merchant Users/Subusers	Chapter 5: *Main Merchant User Actions* and Chapter 6: *Merchant Subuser Actions*

The basic transaction structure is as follows:
`[STX]`identifier**`[FS]`**message**`[ETX]`**

The identifier should be unique to the session, but it may be reused once a response is received from a transaction that shares the same identifier. The identifier may be any unique string of numbers and letters and is echoed in the response Monetra provides.

Both requests and responses share the same formatting requirements. The message portion of the data stream is formatted as detailed below.

If a connection to Monetra is lost while there are pending transactions for your connection, you must issue a Get Unsettled Transactions or Get Failed Transactions report in order to determine the status of your transaction. There is no recovery process to resume a failed session; therefore, a stable connection is strongly recommended.

We suggest using pre-existing free libraries to perform encryption and decryption with TLS/SSL. The most widely used libraries are available at http://www.openssl.org/. These libraries help perform all encryption "behind the scenes".

3.2.2 HTTPS

PCI Notice: Over public networks (like the Internet) or untrusted private networks, PCI Security requires the use of HTTPS instead of HTTP for communication to ensure card numbers and other critical data are not transmitted in plain text.

HTTP is the method of data transfer that web servers use. When you visit a website, that content is delivered to you using the HTTP protocol. HTTPS is simply HTTP tunneled over a TLS/SSL connection. The default port numbers for communication are `8665` for Merchant User actions and `8666` for Admin User actions.

With the goal of keeping it simple, Monetra only supports a small subset of the HTTP protocol. An HTTPS request to Monetra should be formatted as a simple HTTP 1.0 or 1.1 POST command. A Content-Length descriptor is also required to allow the Monetra parser to efficiently identify the start and end of the data being sent. Please verify that the host and port numbers reference your configuration (IP address of Monetra service, and configured ports for HTTPS).

Most programming languages provide HTTP routines to simplify the programming process. You may also wish to look into `curl` as an alternative (http://curl.haxx.se).

HTTP 1.0 is defined in RFC 1945 (http://www.faqs.org/rfcs/rfc1945.html), and HTTP 1.1 is defined in RFC 2616 (http://www.faqs.org/rfcs/rfc2616.html)

3.3 Transaction Structures

The basic protocol used to communicate with Monetra consists of key/value pairs. Beyond that, Monetra can accept transactions formatted as XML or JSON, as outlined below, or sent using the ReST API [https://developers.monetra.com/].

3.3.1 XML Transactions

Each set of XML transactions must begin and end with `<MonetraTrans>` tags. Within these tags is one or more transactions. Each transaction is wrapped in `<Trans>` tags, which must also specify an `identifier` attribute of letters and/or numbers for uniquely identifying that transaction. The sub tags for each transaction are the key/value pairs found in this specification. The order of the key/value data pairs is irrelevant, and Monetra does not return the response key/value pairs in any particular order.

XML responses are encapsulated in `<MonetraResp>` tags, and a corresponding `<Resp>` tag exists for every `<Trans>` tag sent (assuming the structure was initially correct). The `<Resp>` tag has an `identifier` attribute that echoes the identifier sent with each `<Trans>` tag. An additional `<DataTransferStatus>` tag indicating whether or not the request was successfully parsed is always sent with a `code` attribute, which has a value of `SUCCESS` or `FAIL`. On `FAIL`, an error message is enclosed in the `<DataTransferStatus>` tag, as seen in Section 3.3.1.3: XML Error Example.

Every `<Resp>` tag has a `<Code>` tag indicating the overall outcome of the transaction. For comma-separated responses (like you receive when running reports), there is a `<DataBlock>` tag that holds the response data. Otherwise, the response tags are those as defined in this guide.

Note: Some transactional characters might need to be encoded when inserting them into the XML data stream, such as these: `'>'` needs to become `'&gt;'`, `'<'` needs to become `'&lt;'` and `'&'` needs to become `'&amp;'`.

Note: The fields in a `<DataBlock>` tag are guaranteed to not be removed or renamed, but fields can be added or rearranged at any time. Integrations need to parse the header first to find the correct column index for the desired field before reading out the value in that field and should never blindly assume that the data will always be at a specific column index.

3.3.1.1 XML Request Example

Below is an example of an XML request.

```

<MonetraTrans>
 <Trans identifier='1'>
  <username>vitale</username>
  <password>test</password>
  <action>sale</action>
  <account>4012888888881</account>
  <expdate>0512</expdate>
  <amount>12.00</amount>
 </Trans>
 <Trans identifier="can be a string too">
  <username>vitale</username>
  <password>test</password>
  <action>sale</action>
  <account>5454545454545454</account>
  <expdate>0512</expdate>
  <amount>11.00</amount>
 </Trans>
 <Trans identifier='3'>
  <username>vitale</username>
  <password>test</password>
  <action>admin</action>
  <admin>gut</admin>
 </Trans>
</MonetraTrans>
```

3.3.1.2 XML Response Example

Below is an example of an XML response.

```
<MonetraResp>
 <DataTransferStatus code='SUCCESS'/>
 <Resp identifier="can be a string too">
  <code>AUTH</code>
  <verbiage>APPROVAL 123456</verbiage>
  <batch>1</batch>
  <item>1</item>
  <avs>STREET</avs>
  <cv>GOOD</cv>
  <ttid>112</ttid>
 </Resp>
 <Resp identifier='3'>
  <code>SUCCESS</code>
  <DataBlock>
   ttid,type,capture, ...
   1,SALE,1, ...
   5,PREAUTH,0, ...
   7,RETURN,1, ...
  </DataBlock>
 </Resp>
 <Resp identifier='1'>
  <code>DENY</code>
  <verbiage>CVV2 MISMATCH</verbiage>
  <avs>GOOD</avs>
  <cv>BAD</cv>
  <ttid>3842</ttid>
 </Resp>
</MonetraResp>
```

3.3.1.3 XML Error Example

Below is an example of an XML response when the transaction was not structured properly.

```
<MonetraResp>
 <DataTransferStatus code='FAIL'>My descriptive failure reason</DataTransferStatus>
</MonetraResp>
```

3.3.2 JSON Transactions

Each set of JSON transactions must begin with "MonetraTrans". Within this block is one or more transactions. Each transaction begins with a unique identifier of letters and/or numbers. The data for each transactions consists of the key/value pairs found in this specification. The order of the key/value data pairs is irrelevant, and Monetra does not return the response key/value pairs in any particular order.

JSON responses begin with "MonetraResp". Within this block is a transaction response block for each transaction sent. The attribute title of each response block matches the corresponding transaction that was sent. Additionally, a "DataTransferStatus" block is always sent with a code attribute, which has a value of SUCCESS or FAIL. On FAIL, an error message is sent back in the "verbiage" attribute.

Every transaction response block has a "code" attribute indicating the overall outcome of the transaction. For comma-separated responses (like you receive when running reports), there is a "DataBlock" attribute that holds the response data. Otherwise, the response attribute pairs are those as defined in this guide.

Note: The fields in a "DataBlock" tag are guaranteed to not be removed or renamed, but fields can be added or rearranged at any time. Integrations need to parse the header first to find the correct column index for the desired field before reading out the value in that field and should never blindly assume that the data will always be at a specific column index.

3.3.2.1 JSON Request Example

Below is an example of a JSON request.

```
{
 "MonetraTrans": {
  "1": {
   "action": "ping"
  },

  "2": {
   "username": "test_retail:public",
   "password": "publ1ct3st",
   "action": "sale",
   "amount": "2.00",
   "account": "5454545454545454",
   "expdate": "0219",
   "zip": "32606"
  },

  "doesn't have to be a number": {
   "username": "test_retail:public",
   "password": "publ1ct3st",
   "action": "return",
   "amount": "3.00",
   "account": "5454545454545454",
   "expdate": "0219",
   "zip": "32606",
   "cvv": "123"
  },

  "numbers are typically easier than words or sentences": {
   "username": "test_retail:public",
   "password": "publ1ct3st",
   "action": "admin",
   "admin": "gut"
  }
 }
}

```

3.3.2.2 JSON Response Example

Below is an example of a JSON response.

```

{
 "MonetraResp" : {
  "DataTransferStatus" : {
   "code" : "SUCCESS"
  },

  "1" : {
   "verbiage" : "AUTHENTICATION FAILED",
   "msoft_code" : "ACCT_AUTHFAILED",
   "phard_code" : "UNKNOWN",
   "code" : "DENY"
  },

  "2" : {
   "ttid" : "423542",
   "auth" : "214833",
   "timestamp" : "1442607738",
   "phard_code" : "SUCCESS",
   "avs" : "GOOD",
   "verbiage" : "APPROVED",
   "account" : "XXXXXXXXXXXX5454",
   "batch" : "393",
   "code" : "AUTH",
   "pclevel" : "0",
   "rcpt_host_ts" : "091815162218",
   "rcpt_entry_mode" : "M",
   "cardtype" : "MC",
   "item" : "5237",
   "msoft_code" : "INT_SUCCESS"
  },

  "doesn't have to be a number" : {
   "msoft_code" : "INT_SUCCESS",
   "cardtype" : "MC",
   "account" : "XXXXXXXXXXXX5454",
   "phard_code" : "UNKNOWN",
   "verbiage" : "SUCCESS",
   "pclevel" : "0",
   "code" : "AUTH",
   "rcpt_entry_mode" : "M",
   "item" : "5238",
   "timestamp" : "1442607738",
   "rcpt_host_ts" : "091815162218",
   "ttid" : "423543",
   "batch" : "393"
  },

  "numbers are typically easier than words or sentences" : {
   "code" : "SUCCESS",
   "DataBlock" : "ttid,type,capture, ...
    1,SALE,1, ...
    5,PREAUTH,0, ...
    7,RETURN,1, ..."
  }
 }
}

```

3.3.2.3 JSON Error Example

Below is an example of a JSON response when the transaction was not structured properly.

```

{
 "MonetraResp" : {
  "DataTransferStatus" : {
   "code" : "FAIL",
   "verbiage" : "My descriptive failure reason"
  }
 }
}

```

3.3.3 ReST API

For information and details on the ReST API, please see the online documentation at https://developers.monetra.com/.

3.4 Parameter Formatting Legend

The table below represents the message formatting that needs to be sent.

Required Parameter	
Symbol	**Description**
Y	Yes. The parameter is required.
C	Conditional. The requirements of the parameter depend on the presence of other parameters.
O	Optional. The parameter is not required, but sending it will change the scope and behavior of the action and/or the data returned in the response.

Field Formatting	
Symbol	**Description**
A	Alphabetic. Any letter from A to Z. Case does not matter.
N	Numeric. Any digit from 0 to 9.
M	Monetary amount. Any digit from 0 to 9 with optional decimal point and cents, always positive.
S	Special characters. Any character not alphabetic or numeric.
X	Hex data. Pairs of two characters, from 0 to 9 and A to F.

4 Admin User Actions

The following sections in this chapter detail the actions that only Admin Users are allowed to perform.

Please review the required permission level to send Admin User actions. See Section 2.4.1: Admin User Authentication.

4.1 Authenticate As User

As of Monetra 8.0.0, a new feature has been introduced that allows Admin Users with the AUTHAS privilege to run transactions as if they were a Merchant User. Previously, it was common for integrators with this need to perform a getuserinfo request to retrieve the user's password. That feature is no longer supported.

4.1.1 Auth-As Action

The Authenticate-As feature allows any Admin User with the AUTHAS privilege to run a transaction as if they were that user. In the example below, we show how this feature can be used to run a **void** transaction for merchant 'Joe_tractor'.

```
username = MADMIN:Admin_user1
password = MyPa$$WorD
auth-as = Joe_tractor
action = void
ttid = 915576307289978
```

4.1.2 Auth-As Shorthand Notation

The Auth-As feature also allows an abbreviated expression by specifying the Admin User's username separated by a double colon ('::'), followed by the Merchant User's username (and optionally Merchant Subuser) to authenticate as in the username field. This may simplify existing integrations, as an additional key/value pair does not have to be specified. Also, this allows you to Auth-As in the Monetra Client GUI using this shorthand.

The shorthand for authenticating as a Merchant User is as follows:

```
Admin_User[:Admin_Subuser]::Merchant_User[:Merchant_Subuser]
```

```
username = MADMIN:Admin_user1::Joe_tractor
password = MyPa$$WorD
action = void
ttid = 915576307289978
```

4.2 Admin User Management

Subusers allow multiple people to run transactions with varying permission levels. If more than one person will have access to the system, then it is recommended to create subusers that are granted the minimum permissions they require to perform their duties. See Section 2.3.3: Permissions for information on assigning permissions.

4.2.1 Add Admin Subuser

Add an Admin Subuser for administrative tasks.

`Access level:` Section 2.4.1: Admin User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`subuseradd`**
`user`	Y	ANS	Name of user
`pwd`	C	ANS, Max 256	Password to assign to user
`password_expired`	C	A	Boolean flag indicating whether or not to forcibly set the users password as expired. If this is `yes`, then a password change will be enforced at the next login.
`trantypes`	Y	AS	List of transaction permissions, separated by pipes (|), or `all` for all permissions. Valid values are any of the actions in Chapter 4: *Admin User Actions* for Admin users/ subusers or Chapter 6: *Merchant Subuser Actions* for Merchant users/subusers.
`authas_admintypes`	O	AS	List of administrative actions that can be performed using the auth-as feature, separated by pipes (|), or `all` for all actions (the default). Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
`authas_trantypes`	O	AS	List of transaction types that can be performed using the auth-as feature, separated by pipes (|), or `all` for all transactions (the default). Valid values are any of the transaction types listed in Chapter 6: *Merchant Subuser Actions*.

Request Parameters			
Key	**Req**	**Spec**	**Description**
userflags	O	AS	List of flags for the subuser, separated by pipes (|). Possible values: obscure - Always obscure sensitive data unattended - Prevent the password from expiring (useful for connected integrations) selfviewonly - Restrict reports to show only this subuser's data authas - Allow subuser to run transactions as other subusers
obscure	O	A	Boolean flag indicating whether or not to always obscure sensitive data
unattended	O	A	Boolean flag indicating whether or not the account is marked as unattended

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.2.2 Edit Admin Subuser

Edit an Admin Subuser account.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**subuseredit**
user	Y	ANS	Subuser to edit
pwd	C	ANS, Max 256	Password to assign to user
password_expired	C	A	Boolean flag indicating whether or not to forcibly set the users password as expired. If this is yes, then a password change will be enforced at the next login.
trantypes	Y	AS	List of transaction permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 4: *Admin User Actions* for Admin users/ subusers or Chapter 6: *Merchant Subuser Actions* for Merchant users/subusers.
authas_admintypes	O	AS	List of administrative actions that can be performed using the auth-as feature, separated by pipes (|), or all for all actions (the default). Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
authas_trantypes	O	AS	List of transaction types that can be performed using the auth-as feature, separated by pipes (|), or all for all transactions (the default). Valid values are any of the transaction types listed in Chapter 6: *Merchant Subuser Actions*.
userflags	O	AS	List of flags for the subuser, separated by pipes (|). Possible values: obscure - Always obscure sensitive data unattended - Prevent the password from expiring (useful for connected integrations) selfviewonly - Restrict reports to show only this subuser's data authas - Allow subuser to run transactions as other subusers
obscure	O	A	Boolean flag indicating whether or not to always obscure sensitive data
unattended	O	A	Boolean flag indicating whether or not the account is marked as unattended

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.2.3 Delete Admin Subuser

Delete an Admin Subuser account.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**subuserdel**
user	Y	ANS	Name of user

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.2.4 List Admin Subusers

Get a datablock of the Admin Subuser accounts.

Access level: Section 2.4.1: Admin User Authentication Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**subuserlist**
include_subusers	O	A	Boolean flag indicating whether or not to return a list of subusers as well as main users

Report Fields		
Key	**Spec**	**Description**
user	ANS	Name of user
master	A	Boolean flag indicating whether or not the account is a master (main user) account
trantypes	AS	List of transaction permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 4: *Admin User Actions* for Admin users/subusers or Chapter 6: *Merchant Subuser Actions* for Merchant users/subusers.
userflags	AS	List of flags for the subuser, separated by pipes (|). Possible values: obscure - Always obscure sensitive data unattended - Prevent the password from expiring (useful for connected integrations) selfviewonly - Restrict reports to show only this subuser's data authas - Allow subuser to run transactions as other subusers
obscure	A	Boolean flag indicating whether or not to always obscure sensitive data
unattended	A	Boolean flag indicating whether or not the account is marked as unattended
pass_expire_secs	N, Max 20	Number of seconds until the password expires (or -1 for never or 0 for expired)
locked	A	Boolean flag indicating whether or not the account is locked
authas_trantypes	AS	List of transaction types that can be performed using the auth-as feature, separated by pipes (|), or all for all transactions (the default). Valid values are any of the transaction

Report Fields		
Key	**Spec**	**Description**
		types listed in Chapter 6: *Merchant Subuser Actions*.
authas_admintypes	AS	List of administrative actions that can be performed using the auth-as feature, separated by pipes (|), or all for all actions (the default). Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.

4.2.5 Get User Permissions

Get a datablock of the allowed request permissions for the Admin User in question. This request is useful when building applications where you only want to expose specific features for a user for which they have permission to use.

Access level: Section 2.4.1: Admin User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**getperms**

Report Fields		
Key	**Spec**	**Description**
trantypes	AS	List of transaction permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 4: *Admin User Actions* for Admin users/subusers or Chapter 6: *Merchant Subuser Actions* for Merchant users/subusers.
admintypes		Unused
userflags	AS	List of flags for the subuser, separated by pipes (|). Possible values: obscure - Always obscure sensitive data unattended - Prevent the password from expiring (useful for connected integrations) selfviewonly - Restrict reports to show only this subuser's data authas - Allow subuser to run transactions as other subusers
obscure	A	Boolean flag indicating whether or not to always obscure sensitive data

Report Fields		
Key	**Spec**	**Description**
unattended	A	Boolean flag indicating whether or not the account is marked as unattended
authas_trantypes	AS	List of transaction types that can be performed using the auth-as feature, separated by pipes (|), or all for all transactions (the default). Valid values are any of the transaction types listed in Chapter 6: *Merchant Subuser Actions*.
authas_admintypes	AS	List of administrative actions that can be performed using the auth-as feature, separated by pipes (|), or all for all actions (the default). Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.

4.2.6 Check Admin User Password

Verify the current Admin User's password.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**chkpwd**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
pass_expire_secs	N, Max 20	Number of seconds until the password expires (or -1 for never or 0 for expired)

4.2.7 Change Admin User Password

Change the Admin User's password.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**chngpwd**
pwd	Y	ANS, Max 256	Password to assign to user

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.3 Merchant User Management

Merchant User management is one of the most critical aspects of properly configuring and operating a Monetra system. In particular, ensuring the Merchant User's profile information is correct is especially important. See Section 2.3.3: Permissions for information on assigning permissions.

4.3.1 Add Merchant User

Add a new merchant account, which is tied to a Merchant User.

Because you will be creating a new merchant account, it is important to have all the required information ready before issuing this request. To prepare the processor information, it is recommended that you run a `proclist` request to obtain a list of available processors (with capabilities), select the processor that will be used for the merchant account from that list, and then issue a `procfields` request to return the required data elements and formatting information that must be passed with the transaction.

This request can also be used to create a new subaccount on an existing account. In this situation, you would include the parameter `sub` and take care that the routes do not overlap. For more information, see Section 4.3.10: List Subaccounts.

Note: Monetra will DENY the request if all required fields are not present and formatted correctly.

Note: It is important that you get all the information provided by the merchant acquirer (in the var setup worksheet) into Monetra exactly as provided. You should also pay attention to the configuration of the account with settings such as industry type, `fraudautodeny`, etc.

`Access level:` Section 2.4.1: Admin User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`adduser`**
`user`	Y	ANS	Name of user
`pwd`	Y	ANS, Max 256	Password to assign to user
`sub`	O	N, Max 10	Subaccount ID (for split routes), or `0` for the default route
`proc`	Y	A	Processing institution
`indcode`	Y	A, Max 2	Industry code for processing transactions. See Appendix C.5: Industry Codes.

Request Parameters			
Key	**Req**	**Spec**	**Description**
`cardtypes`	Y	AS	List of card types, separated by pipes (|) or pluses (+). See Appendix C.1: Card Types.
`mode`	Y	A	Supported routes for sending transactions online. Possible values: `AUTH` - Authorization route only `SETTLE` - Settlement route only `BOTH` - Both authorization and settlement
`email`	O	ANS	Email address for sending notices/cron emails
`conn_priority`	O	AS	Shows the connection priority
`fraudautodeny`	O	AS	List of rules used to automatically deny transactions based on fraud checks, separated by pipes (|). See Appendix B.2.1: AVS Result Codes. Possible values: `deny_never` - Never auto deny `deny_avszip` - Auto deny if zip does not match `deny_avsstreet` - Auto deny if street does not match `deny_cv` - Auto deny if CV does not match `deny_noauto` - Auto deny checks will only take place if `fraudautodeny`=`yes` is sent with transaction
`merch_name`	O	ANS	Merchant's name
`merch_descr`	O	ANS	Description to identify Merchant
`merch_addr1`	O	ANS	Merchant's address - line 1
`merch_addr2`	O	ANS	Merchant's address - line 2
`merch_addr3`	O	ANS	Merchant's address - line 3
`merch_phone`	O	ANS	Merchant's phone number
`merch_email`	O	ANS	Merchant's email
`merch_url`	O	ANS	Merchant's website URL
`merch_lang`	O	A	2-character language code. Possible values: `en` - English `fr` - French `es` - Spanish `de` - German `it` - Italian

Request Parameters			
Key	**Req**	**Spec**	**Description**
merch_tz	O	ANS	Merchant's time zone. See Section 4.5.4: List Time Zones.
merch_cashback	O	ANS	List of dollar amounts to display on screen during cash back prompting, separated by pipes (|). All values are whole dollar amounts, and the letter O can be used to indicate the ability to enter a custom amount. For example, 5|10|25|O would prompt for $5, $10, $25, or Other.
merch_cashbackmax	O	M	Maximum cash back amount
merch_cashback_purchmin	O	M	Minimum cash back purchase amount
merch_tippercent	O	M	Merchant tip percentages to display
merch_msr_nosig_limit	O	M	Limit to not require a signature for MSR
merch_enc_provider	O	A	Processor-level, end-to-end encryption provider. Currently, only a value of bluefin is supported. If used, encryptedonly must be set in merch_flags.
merch_pushnotification_id	O	ANS	Registered push notification ID, as returned from a pushnotification=add request. See Section 4.12: Push Notification Management.
merch_flags	O	ANS	Merchant flags for modifying internal processing. Possible values: account_updater - Mark the merchant's tokens as eligible for Account Updater. This is not valid if the Merchant User is part of token group. allow_invoice_sms - Allow Merchant User and its Subusers to send SMS messages for invoices. allow_invoicing - Allow Merchant User and its Subusers to create invoices and products. debit_standin_allowed - Allow stand-in processing of Debit transactions in UniTerm. emvpin_standin_allowed - Allow stand-in processing of EMV Online PIN in UniTerm. emvsuppresssig - Suppress signature prompting in UniTerm.

Request Parameters			
Key	Req	Spec	Description
			encryptedonly - Only allow encrypted transactions. orders_req_level3 - Require Level III data on Retail, Ecomm, and MOTO orders. Must have merch_l3_commoditycode and merch_l3_productcode set. prepop_level3 - Prepopulate a single line item on Level III-eligible transactions that do not already have any line items. This is done at settlement time using the default values set in merch_l3_commoditycode, merch_l3_description, and merch_l3_productcode. Using this flag might require sign-off by merchant's internal risk department. prepop_tax - Prepopulate a tax amount on transactions with no provided tax. The amount is reverse-calculated based on merch_tax_rate and subtracts any existing tip amount, duty, or freight before performing the calculation. ratequal_nontax - For non-taxable transactions that can't qualify for Level II rates but are using a business/corporate/ purchase card, modify data internally to force best rate qualification. This works by claiming a tax rate of 0.1% + 0.1 to the card brands but hiding the internal workings to the merchant and customer. Using this flag might require sign-off by merchant's internal risk department. receive_receipts_detail - Receive merchant receipt copies by email for trandetail requests. receive_receipts_invoice - Receive merchant receipt copies by email for paid invoices receive_receipts_recurring - Receive merchant receipt copies by email for recurring transactions. recurring_skipmissed - When a recurring payment has missed payments, do not re-attempt the missed payments when the schedule is active again.

Request Parameters			
Key	**Req**	**Spec**	**Description**
`merch_l3_commoditycode`	O	AN	Level III Item Commodity Code. International description code of the individual good or service being supplied. Required if `prepop_level3` is set in `merch_flags`.
`merch_l3_description`	O	ANS	Item Descriptor. Merchant-defined description of the item or service being sold. Required if `prepop_level3` is set in `merch_flags`.
`merch_l3_productcode`	O	ANS	Product Code. Merchant-defined code for the item being purchased such as a UPC number, SKU, or Item Number. Required if `prepop_level3` is set in `merch_flags`.
`merch_tax_rate`	O	M	Tax rate as a percentage between 0.1 and 49.99. Required if `prepop_tax` is set in `merch_flags`.
`merch_token_group`	O	ANS	Token group to add the merchant to. See Section 4.10.1: Manage Token Groups.
`merch_custom_fields`	O	ANS	List of custom field names to allow for specifying custom fields sent with a transaction, separated by commas (`,`). The field names can have the character set of 'a-z0-9_-' and be up to 32 characters each. * During each transaction request (e.g. 'sale', 'preauth', 'return', 'force'), one or more of the configured merchant custom fields may be specified using the standard key/value pair format as defined by the protocol. This data will be stored with the transaction record. * Merchant custom fields can be updated during 'fieldedit', 'adjust', 'preauthcomplete', or 'capture' requests. * For each merchant custom field configured for a merchant, a new column will appear in the **`gut`**, **`gl`**, and **`gft`** reports and will contain any custom data presented with each transaction.
`merch_req_fields`	O	AS	List of field names to require for every transaction, separated by commas (`,`). Possible values: `tax`

Request Parameters			
Key	**Req**	**Spec**	**Description**
			examount custref ordernum stationid clerkid comments cardholdername descmerch descloc cvv2 (only required if online and not EMV or Swipe) zip (only required if online and not EMV or Swipe) any of the 'merch_custom_fields'
merch_customer_fields	O	ANS	List of custom field names to allow when adding/editing customers, separated by commas (,). The field names can have the character set of 'a-z0-9_-' and be up to 32 characters each.
emv_entrymodes	O	AS	List of EMV entry modes, separated by pipes (|). See Appendix C.2: EMV Entry Modes.
emv_termcaps	O	AS	List of EMV terminal capabilities, separated by pipes (|). These must match the capabilities of the device in the certification list documented in the UniTerm guide. See Appendix C.3: EMV Terminal Capabilities.
emv_testmode	O	A	Boolean flag indicating whether or not to load and verify only test CAPKs. Defaults to no, meaning to load only production CAPKs.
emv_contact_nocvm_limit	O	M	EMV Contact NoCVM Limit
emv_ctls_nocvm_limit	O	M	EMV Contactless NoCVM Limit
txn_export_key	O	ANS	Export key, in this format: ID returned from cardshieldprovision request followed by pipe separator (|) followed by RSA public key data, including standard start and end sentinels (i.e., -----{BEGIN|END} PUBLIC KEY-----) and keeping newlines intact

Request Parameters			
Key	**Req**	**Spec**	**Description**
In addition to the fields above, you must also populate processor-specific fields, which can vary from processor to processor. See note above about using proclist and procfields to determine what is required for your particular processor.			

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.3.2 Edit Merchant User

Edit an existing merchant account. When you edit a field, the new value will overwrite and replace the old value.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**edituser**
user	Y	ANS	Name of user
pwd	Y	ANS, Max 256	Password to assign to user
proc	Y	A	Processing institution
indcode	Y	A, Max 2	Industry code for processing transactions. See Appendix C.5: Industry Codes.
cardtypes	Y	AS	List of card types, separated by pipes (|) or pluses (+). See Appendix C.1: Card Types.
mode	Y	A	Supported routes for sending transactions online. Possible values: AUTH - Authorization route only

Request Parameters			
Key	**Req**	**Spec**	**Description**
			`SETTLE` - Settlement route only `BOTH` - Both authorization and settlement
`email`	O	ANS	Email address for sending notices/cron emails
`conn_priority`	O	AS	Shows the connection priority
`fraudautodeny`	O	AS	List of rules used to automatically deny transactions based on fraud checks, separated by pipes (`|`). See Appendix B.2.1: AVS Result Codes. Possible values: `deny_never` - Never auto deny `deny_avszip` - Auto deny if zip does not match `deny_avsstreet` - Auto deny if street does not match `deny_cv` - Auto deny if CV does not match `deny_noauto` - Auto deny checks will only take place if `fraudautodeny=yes` is sent with transaction
`merch_name`	O	ANS	Merchant's name
`merch_descr`	O	ANS	Description to identify Merchant
`merch_addr1`	O	ANS	Merchant's address - line 1
`merch_addr2`	O	ANS	Merchant's address - line 2
`merch_addr3`	O	ANS	Merchant's address - line 3
`merch_phone`	O	ANS	Merchant's phone number
`merch_email`	O	ANS	Merchant's email
`merch_url`	O	ANS	Merchant's website URL
`merch_lang`	O	A	2-character language code. Possible values: `en` - English `fr` - French `es` - Spanish `de` - German `it` - Italian
`merch_tz`	O	ANS	Merchant's time zone. See Section 4.5.4: List Time Zones.
`merch_cashback`	O	ANS	List of dollar amounts to display on screen during cash back prompting, separated by pipes (`|`). All values are whole dollar amounts, and the letter `O` can be used

Request Parameters			
Key	Req	Spec	Description
			to indicate the ability to enter a custom amount. For example, 5|10|25|O would prompt for $5, $10, $25, or Other.
merch_cashbackmax	O	M	Maximum cash back amount
merch_cashback_purchmin	O	M	Minimum cash back purchase amount
merch_tippercent	O	M	Merchant tip percentages to display
merch_msr_nosig_limit	O	M	Limit to not require a signature for MSR
merch_enc_provider	O	A	Processor-level, end-to-end encryption provider. Currently, only a value of bluefin is supported. If used, encryptedonly must be set in merch_flags.
merch_pushnotification_id	O	ANS	Registered push notification ID, as returned from a pushnotification=add request. See Section 4.12: Push Notification Management.
merch_flags	O	ANS	Merchant flags for modifying internal processing. Possible values: account_updater - Mark the merchant's tokens as eligible for Account Updater. This is not valid if the Merchant User is part of token group. allow_invoice_sms - Allow Merchant User and its Subusers to send SMS messages for invoices. allow_invoicing - Allow Merchant User and its Subusers to create invoices and products. debit_standin_allowed - Allow stand-in processing of Debit transactions in UniTerm. emvpin_standin_allowed - Allow stand-in processing of EMV Online PIN in UniTerm. emvsuppresssig - Suppress signature prompting in UniTerm. encryptedonly - Only allow encrypted transactions. orders_req_level3 - Require Level III data on Retail, Ecomm, and MOTO orders. Must have merch_l3_commoditycode and merch_l3_productcode set.

Request Parameters			
Key	Req	Spec	Description
			prepop_level3 - Prepopulate a single line item on Level III-eligible transactions that do not already have any line items. This is done at settlement time using the default values set in merch_l3_commoditycode, merch_l3_description, and merch_l3_productcode. Using this flag might require sign-off by merchant's internal risk department. prepop_tax - Prepopulate a tax amount on transactions with no provided tax. The amount is reverse-calculated based on merch_tax_rate and subtracts any existing tip amount, duty, or freight before performing the calculation. ratequal_nontax - For non-taxable transactions that can't qualify for Level II rates but are using a business/corporate/ purchase card, modify data internally to force best rate qualification. This works by claiming a tax rate of 0.1% + 0.1 to the card brands but hiding the internal workings to the merchant and customer. Using this flag might require sign-off by merchant's internal risk department. receive_receipts_detail - Receive merchant receipt copies by email for trandetail requests. receive_receipts_invoice - Receive merchant receipt copies by email for paid invoices receive_receipts_recurring - Receive merchant receipt copies by email for recurring transactions. recurring_skipmissed - When a recurring payment has missed payments, do not re-attempt the missed payments when the schedule is active again.
merch_l3_commoditycode	O	AN	Level III Item Commodity Code. International description code of the individual good or service being supplied. Required if prepop_level3 is set in merch_flags.

Request Parameters			
Key	**Req**	**Spec**	**Description**
`merch_l3_description`	O	ANS	Item Descriptor. Merchant-defined description of the item or service being sold. Required if `prepop_level3` is set in `merch_flags`.
`merch_l3_productcode`	O	ANS	Product Code. Merchant-defined code for the item being purchased such as a UPC number, SKU, or Item Number. Required if `prepop_level3` is set in `merch_flags`.
`merch_tax_rate`	O	M	Tax rate as a percentage between 0.1 and 49.99. Required if `prepop_tax` is set in `merch_flags`.
`merch_token_group`	O	ANS	Token group to add the merchant to. See Section 4.10.1: Manage Token Groups.
`merch_custom_fields`	O	ANS	List of custom field names to allow for specifying custom fields sent with a transaction, separated by commas (`,`). The field names can have the character set of 'a-z0-9_-' and be up to 32 characters each. * During each transaction request (e.g. 'sale', 'preauth', 'return', 'force'), one or more of the configured merchant custom fields may be specified using the standard key/value pair format as defined by the protocol. This data will be stored with the transaction record. * Merchant custom fields can be updated during 'fieldedit', 'adjust', 'preauthcomplete', or 'capture' requests. * For each merchant custom field configured for a merchant, a new column will appear in the **`gut`**, **`gl`**, and **`gft`** reports and will contain any custom data presented with each transaction.

`merch_req_fields`	O	AS	List of field names to require for every transaction, separated by commas (`,`). Possible values:
			`tax`
			`examount`
			`custref`
			`ordernum`
			`stationid`
			`clerkid`

Request Parameters			
Key	Req	Spec	Description
			comments cardholdername descmerch descloc cvv2 (only required if online and not EMV or Swipe) zip (only required if online and not EMV or Swipe) any of the 'merch_custom_fields'
merch_customer_fields	O	ANS	List of custom field names to allow when adding/editing customers, separated by commas (`,`). The field names can have the character set of 'a-z0-9_-' and be up to 32 characters each.
emv_entrymodes	O	AS	List of EMV entry modes, separated by pipes (`|`). See Appendix C.2: EMV Entry Modes.
emv_termcaps	O	AS	List of EMV terminal capabilities, separated by pipes (`|`). These must match the capabilities of the device in the certification list documented in the UniTerm guide. See Appendix C.3: EMV Terminal Capabilities.
emv_testmode	O	A	Boolean flag indicating whether or not to load and verify only test CAPKs. Defaults to `no`, meaning to load only production CAPKs.
emv_contact_nocvm_limit	O	M	EMV Contact NoCVM Limit
emv_ctls_nocvm_limit	O	M	EMV Contactless NoCVM Limit
txn_export_key	O	ANS	Export key, in this format: ID returned from cardshieldprovision request followed by pipe separator (`|`) followed by RSA public key data, including standard start and end sentinels (i.e., `-----{BEGIN|END} PUBLIC KEY-----`) and keeping newlines intact
In addition to the fields above, you may also edit processor-specific fields, which can vary from processor to processor. See the adduser request above to determine what was required for your particular processor.			

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.3.3 Delete Merchant User

Delete a merchant account and its associated Merchant User from Monetra.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**deluser**
user	Y	ANS	Name of user

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.3.4 List Merchant Users

Get a datablock of Merchant Users, optionally limited to a single Merchant User, along with some information about each

Access level: Section 2.4.1: Admin User Authentication Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**listusers**
user	O	ANS	Name of user
include_subusers	O	A	Boolean flag indicating whether or not to return a list of subusers as well as main users

Report Fields		
Key	**Spec**	**Description**
userlist	ANS	Name of user
user status	A	Status of user. Possible values: E - Enabled D - Disabled
master	A	Boolean flag indicating whether or not the account is a master (main user) account
trantypes	AS	For subusers, list of transaction permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 4: *Admin User Actions* for Admin Subusers or Chapter 6: *Merchant Subuser Actions* for Merchant Subusers.
admintypes	AS	For subusers, list of administrative permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
profile_id	N, Max 20	Profile ID (or 0 if not set)
pass_expire_secs	N, Max 20	Number of seconds until the password expires (or -1 for never or 0 for expired)
merch_name	ANS	Merchant's name
email	ANS	Email address for sending notices/cron emails
merch_descr	ANS	Description to identify Merchant
locked	A	Boolean flag indicating whether or not the account is locked

4.3.5 Get Merchant User Information

Get information for the provided Merchant User.

`Access level:` Section 2.4.1: Admin User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**getuserinfo**
user	Y	ANS	Name of user

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
cardtypes	AS	List of card types, separated by pipes (|) or pluses (+). See Appendix C.1: Card Types.
countrycode	N, Fixed 3	A reference code to the country the Merchant is processing in. (ISO 3166-1 numeric, ex: USA = 840).
currencycode	N, 3	A reference code to the currency the Merchant is processing in. (ISO 4217, ex: USA = 840).
emv_entrymodes	AS	List of EMV entry modes, separated by pipes (|). See Appendix C.2: EMV Entry Modes.
emv_termcaps	AS	List of EMV terminal capabilities, separated by pipes (|). These must match the capabilities of the device in the certification list documented in the UniTerm guide. See Appendix C.3: EMV Terminal Capabilities.
emv_testmode	A	Boolean flag indicating whether or not to load and verify only test CAPKs. Defaults to no, meaning to load only production CAPKs.
indcode	A, Max 2	Industry code for processing transactions. See Appendix C.5: Industry Codes.

Response Parameters		
Key	**Spec**	**Description**
merch_lang	A	2-character language code. Possible values: en - English fr - French es - Spanish de - German it - Italian
merch_pushnotification_id	ANS	Registered push notification ID, as returned from a pushnotification=add request. See Section 4.12: Push Notification Management.
mode	A	Supported routes for sending transactions online. Possible values: AUTH - Authorization route only SETTLE - Settlement route only BOTH - Both authorization and settlement
proc	A	Processing institution
sub	N, Max 10	Subaccount ID (for split routes), or 0 for the default route
zipcode	AN	The zip code for the retail location of the Merchant or the Merchant's headquarters. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
In addition to the parameters above, this request returns parameters that are specific to the selected processor.		

4.3.6 Get Merchant User Count

Get the current count of Merchant Users as well as the maximum allowed for the license.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**listusers**
listusers	Y		**count**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
user_count	N	Current number of Merchant Users
user_limit	N	Maximum number of Merchant Users allowed with this license, or unlimited for no limit.

4.3.7 Disable Merchant User

Disable a merchant account and its associated Merchant User without removing it from the system.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**disableuser**
user	Y	ANS	Name of user

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.3.8 Enable Merchant User

Re-enable a previously disabled user account.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**enableuser**
user	Y	ANS	Name of user

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.3.9 Unlock Merchant User

Unlock a merchant account and its associated Merchant User.

A merchant account can be locked due to exceeding the limit of password tries. See the entry for `password_lockout_seconds` in the Secure Install Guide [https://www.monetra.com/documentation] for more information on password lockouts.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**unlockacct**
user	Y	ANS	Name of user

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.3.10 List Subaccounts

Get a datablock of the subaccounts for a particular user. This is mainly used for split routing. See *Adding Additional Routes* in the Secure Install Guide [https://www.monetra.com/documentation] for more information on splitting routes.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**getsubaccts**
user	Y	ANS	Name of user

Report Fields		
Key	**Spec**	**Description**
sub	N, Max 10	Subaccount ID (for split routes), or 0 for the default route

4.3.11 List Statistics

Get a datablock of the completed/settled transaction statistics for one or all system users.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**liststats**
bdate	C	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	C	ANS	Ending date. If not specified, assume infinite. See Appendix C.8: Date Formats.
user	O	ANS	Name of user. If not sent, applies to all users.
totalsonly	O	A	Boolean flag indicating whether or not to output only the `totals` line. Defaults to `no`.

Report Fields		
Key	**Spec**	**Description**
user	ANS	Name of user
totaltransNum	N	Total number of authorizations and returns
totaltransAmount	M	Aggregate authorization amounts minus aggregate return amounts
totalAuthNum	N	Total number of authorizations
totalAuthAmount	M	Aggregate amount of all authorizations
totalReturnNum	N	Total number of returns
totalReturnAmount	M	Aggregate amount of all returns

4.4 Processor Management

Processor management is used for both system and user administration.

4.4.1 List Processors

Get a datablock of loaded processors, whether or not they are active, and their capabilities.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**proclist**
proc	O	A	Processing institution

Report Fields

Key	Spec	Description
proc	A	Processing institution
active	A	Boolean flag indicating whether or not the account is currently active
cardtypes	AS	List of card types, separated by pipes (|) or pluses (+). See Appendix C.1: Card Types.
emv_entrymodes	AS	List of EMV entry modes, separated by pipes (|). See Appendix C.2: EMV Entry Modes.
credit_trantypes	AS	List of transaction types supported for credit card, separated by pipes (|). See Section 6.1: Credit Card and Debit Card Transactions.
debit_trantypes	AS	List of transaction types supported for debit cards, separated by pipes (|). See Section 6.1: Credit Card and Debit Card Transactions.
ebt_trantypes	AS	List of transaction types supported for EBT cards, separated by pipes (|). See Section 6.2: EBT Transactions.
gift_trantypes	AS	List of transaction types supported for gift cards, separated by pipes (|). See Section 6.3: Gift Card Transactions.
check_trantypes	AS	List of transaction types supported for checks and ACH, separated by pipes (|). See Section 6.4: Check Transactions.
supported_conns	AS	List of connectivity methods for processor, separated by pipes (|). See Appendix C.6: Connectivity Modes.
supported_modes	A	Supported routes for sending transactions online. Possible values: AUTH - Authorization route only SETTLE - Settlement route only BOTH - Both authorization and settlement
supported_industries	ANS	List of supported industries, separated by pipes (|). See Appendix C.5: Industry Codes.
displayname	ANS	Human-readable display name
helpdesk_phone	ANS	Contact information for the processor (if available)
version	ANS	Module version number
features	ANS	List of processor features, separated by pipes (|). See Appendix C.4: Processor Features.

4.4.2 List Processor Fields

Get a datablock of settings for the processor module.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**procfields**
proc	Y	A	Processing institution

Report Fields		
Key	**Spec**	**Description**
name	ANS	Long-form name
req_credit	A	Boolean flag indicating whether or not the setting is required when credit cards are enabled
req_debit	A	Boolean flag indicating whether or not the setting is required when debit cards are enabled
req_gift	A	Boolean flag indicating whether or not the setting is required when gift cards are enabled
req_ebt	A	Boolean flag indicating whether or not the setting is required when EBT cards are enabled
req_dial	A	Boolean flag indicating whether or not the setting is required when dial-up connectivity is enabled
req_ip	A	Boolean flag indicating whether or not the setting is required when IP connectivity is enabled
req_https	A	Boolean flag indicating whether or not the setting is required when HTTPS connectivity is enabled
req_ssl	A	Boolean flag indicating whether or not the setting is required when SSL connectivity is enabled
req_other	A	Boolean flag indicating whether or not the setting is required when *other* connectivity is enabled
req_auth	A	Boolean flag indicating whether or not the setting is required for authorization routes

Report Fields		
Key	**Spec**	**Description**
`req_settle`	A	Boolean flag indicating whether or not the setting is required when for settlement routes
`len_min`	N	Minimum value length
`len_max`	N	Maximum value length

`field_type`	A	Allowed characters in value. Possible values:
		`VAR_TYPE_ALPHA_ONLY` - Letters only
		`VAR_TYPE_ALPHA_CAPS_ONLY` - Uppercase letters only
		`VAR_TYPE_ALPHA_LOWER_ONLY` - Lowercase letters only
		`VAR_TYPE_NUM_ONLY` - Numbers only
		`VAR_TYPE_NUM_W_DASH_ONLY` - Numbers only, with dashes (`-`) allowed
		`VAR_TYPE_NUM_W_DEC_ONLY` - Numbers only, with decimal points (`.`) allowed
		`VAR_TYPE_ALPHANUM` - Letters and numbers
		`VAR_TYPE_BOOLEAN_DEF_TRUE` - Boolean values (`yes`/`no`,`true`/`false`) only, defaults to `true`
		`VAR_TYPE_BOOLEAN_DEF_FALSE` - Boolean values (`yes`/`no`,`true`/`false`) only, defaults to `false`
		`VAR_TYPE_BINRANGE` - BIN range characters only
		`VAR_TYPE_ANY` - No restrictions

`submitteronly`	A	Boolean flag indicating whether or not the setting is exclusively for Big Batch Admin Users. See Section 4.11: Big Batch Aggregation.
`displayname`	ANS	Human-readable display name
`desc`	ANS	Human-readable description

4.4.3 Show Processor Status

Get the configuration of a processor's connection.

`Access level:` Section 2.4.1: Admin User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**procstatus**
proc	Y	A	Processing institution

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
configuredmethods	AS	List of connection methods for processor, ordered by priority, separated by semicolons (;). See Appendix C.6: Connectivity Modes.
consecerrors	N, Max 20	Consecutive error (connectivity) count for processor/method
currentmethod	A	The current default connectivity method to use for the processor, takes into account offline methods
offlinemethods	ANS	List of values for offline connectivity for processor, ordered by priority, separated by semicolons (;). Each set consists of two values: the connectivity method and the time in seconds for how long the method has been offline, both separated by a colon (:). Example: `ssl:60;https:30`. See Appendix C.6: Connectivity Modes.
onlinemethods	AS	List of online connectivity methods for processor, ordered by priority, separated by semicolons (;). See Appendix C.6: Connectivity Modes.

4.5 System Information

These requests return a variety of information, depending on the value of the `sysinfo` parameter.

4.5.1 List Country Codes

Get a datablock of supported country codes.

Access level:	Section 2.4.1: Admin User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		sysinfo
sysinfo	Y		countrycodes

Report Fields		
Key	**Spec**	**Description**
name	ANS	Name of country
iso_alpha3	A, Fixed 3	Alphabetic country code (ISO 3166-1 alpha-3)
iso_numeric	N, Fixed 3	Numeric country code (ISO 3166-1 numeric)
iana	ANS	IANA top-level domain

4.5.2 List Currency Codes

Get a datablock of supported currency codes.

Access level:	Section 2.4.1: Admin User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		sysinfo
sysinfo	Y		currencycodes

Report Fields		
Key	**Spec**	**Description**
name	ANS	Long-form name
iso_alpha	A, Fixed 3	Alphabetic currency code (ISO 4217 alpha-3)
iso_numeric	N, Fixed 3	Numeric currency code (ISO 4217 numeric)
exponent	N	Number of digits in exponent

4.5.3 Get License Information

Get detailed Monetra license information, such as transaction limits, company name, restrictions, etc.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**sysinfo**
sysinfo	Y		**licinfo**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
admincount	N, Max 10	The number of Monetra Administrator applications that can simultaneously connect to the Monetra Server
annual_trans_limit	N, Max 10	Number of transactions allowed per year, or '0' for unlimited. If the Admin User is configured with an email, warnings will be sent out when the number of transactions reaches 75, 85, and 95 percent of this limit.

Response Parameters

Key	Spec	Description
		Note: The progress towards the limit is a sliding window and is not calculated on an annual basis. For more information, see this FAQ. [https://www.monetra.com/faqs/license-details/transaction-throughput]
cardshielddevices	N, Max 10	The number of Encrypting Devices that can simultaneously connect to the Monetra Server
clientcount	N, Max 10	The number of Monetra Client applications that can simultaneously connect to the Monetra Server
compname	ANS	Name of Company associated with Monetra license
flags	ANS	Various license configurations and add-ons. Possible values: • DEMO - Demo license • EXTSQL - Alternative SQL databases supported (that is, other than the default SQLite). See https://www.monetra.com/faqs/license-details/sql-database-choices. • UNSIGNEDMODS - Custom (unsigned) modules supported. See https://www.monetra.com/faqs/license-details/custom-modules. • REMOTEIP - Leased line support. • HSM - Use of an HSM for data encryption supported. See https://www.monetra.com/faqs/security/does-monetra-support-hsms. • LEVEL3 - Level 3 transactions supported. See Section 5.8: Level III. • EMV - EMV (chip card) transaction supported. See https://www.monetra.com/emv. • BIGBATCH - Big Batch configuration supported. See Section 4.11: Big Batch Aggregation. • INTERCHANGE - Transaction responses include low-level details from Monetra. • TRANSAFE - Stand-in transactions supported.
license_id	N, Max 10	Unique license identifier

Response Parameters		
Key	**Spec**	**Description**
licensecount	N, Max 10	The number of Monetra instances allowed to run in a cluster
max_version	AS	Maximum Monetra version this license supports
merchant_restrictions	AS	Any restrictions that may exist on what processing institution or merchant account numbers may be used
num_users	N, Max 10	Number of Merchant accounts allowed (0=unlimited)
os	ANS	The Operating System the license is provisioned for. See https://www.monetra.com/faqs/license-details/operating-system-choices.
partner_id	N, Max 10	Partner ID who sold the license
required_modules	AS	Semicolon-separated list of modules which must be loaded
unitermcount	N, Max 10	The number of UniTerm Devices that can simultaneously connect to the Monetra Server

4.5.4 List Time Zones

Get a datablock of allowed time zones.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**sysinfo**
sysinfo	Y		**timezones**

Report Fields		
Key	**Spec**	**Description**
timezone	AS	Time zone name

4.5.5 Get Transaction Types

Get a list of the various actions that different levels of users can perform.

A user/subuser can perform only the actions for which they have explicit permissions. For example, If an Admin User/Subuser needs to create a new Merchant User, then they need to have the ADDUSER permission in madmintypes. Similarly, if a Merchant User/Subuser wants to run a Return, then they need the permission RETURN in trantypes.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**sysinfo**
sysinfo	Y		**trantypes**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
madmintypes	AS	List of MADMIN permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 4: *Admin User Actions*.
admintypes	AS	List of administrative permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
trantypes	AS	List of transaction permissions, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 4: *Admin User Actions* for Admin users/subusers or Chapter 6: *Merchant Subuser Actions* for Merchant users/subusers.

4.5.6 List Versions

Get a datablock of the versions for Monetra, the database schema, and the loaded modules, all in these formats:

- module name
- display name
- module type
- module version

Access level: Section 2.4.1: Admin User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**sysinfo**
sysinfo	Y		**versions**

Report Fields		
Key	**Spec**	**Description**
name	ANS	Module name
displayname	ANS	Human-readable display name
type	ANS	Module type. Possible values: LOGICAL MODULE - Sets up and manages transactions CONNECTION MODULE - Handles inbound/ outbound connections PROTOCOL MODULE - Communicates with processing institutions CRYPTO MODULE - Cryptographic module ENCRYPTION PROVIDER MODULE - Communicates with external card encryption providers ACCOUNT UPDATER MODULE - Communicates with external account update providers
version	ANS	Module version number

4.6 System Logging

4.6.1 Log Levels

The table below outlines all of the available log levels and identifies which ones are considered PCI compliant and which are not. As per PCI standards, non-PCI-compliant levels cannot be logged to disk. If you attempt to set the logging level with any non-PCI-compliant flag, Monetra will reject the request. To log non-PCI-compliant levels, see Section 4.6.3: Write log to Buffer.

Note: If the `obscure` flag is set for the user, only the PCI-compliant levels are allowed.

Level	Compliance	Description
`INIT`	PCI	Basic initialization info, such as Monetra version
`CONF`	PCI	Show configuration and startup details
`WARN`	PCI	Warnings such as misconfiguration, etc.
`INFO`	PCI	Un-categorized short information (like stats)
`TRAN`	PCI	Basic information on when a transaction enters the queue and is finished
`ERROR`	PCI	Any significant error condition
`CRIT`	PCI	A critical/significant error which must not be ignored
`TRAN_DETAIL`	PCI	Basic incoming parameters as parsed (with sensitive data sanitized)
`CONN`	PCI	Log connection details such as IP address, when it is opened/closed/etc.
`PROC`	PCI	Log connection info to processors and details when transacting
`PROC_DETAIL`	PCI	Log more detailed, sanitized, information such as the traces
`TRAN_TRACE`	Non-PCI	Parsed incoming and outgoing transaction requests un-obscured
`TRACE_IN`	Non-PCI	Raw trace of data in and out from client connections
`TRACE_OUT`	Non-PCI	Raw trace of data between Monetra and the processors
`SQL`	Non-PCI	Raw SQL statements
`DEBUG`	Non-PCI	Very verbose, un-grouped data
`DEV`	Non-PCI	Reserved for internal development use only
`PROC_TRACE`	Non-PCI	Non-sanitized version of PROC_DETAIL

4.6.2 Set Logging Levels

Set the current log output levels for Monetra.

This overrides the log levels that were set in main.conf at startup. The changes persist until shutdown (or until this action is run again).

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**setlogging**
debug	Y	AS	List of log levels, separated by pipes (|). For available levels, see Section 4.6.1: Log Levels.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.6.3 Write log to Buffer

Write Monetra log to an in-memory buffer.

The only way to read a log with levels that are out of PCI compliance is by using an in-memory buffer. This allows for temporary logging without changing the log levels written to disk. To retrieve the log, see Section 4.6.4: Retrieve Log Buffer.

Note: Default in-memory buffer size is 1 MB, and buffers are only available for a maximum of 10 minutes.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`setlogging`**
`setlogging`	Y		**`buffer`**
`debug`	O	AS	List of log levels, separated by pipes (`|`). For available levels, see Section 4.6.1: Log Levels.
`buffer_size`	O	N	Maximum size of the buffer, in bytes (default 1 MB)
`bufferid`	O	A	Boolean flag indicating whether or not a unique ID should be returned for this buffer. This ID is later used with a Retrieve Log Buffer request to retrieve that specific buffer. Otherwise, only one buffer (the default buffer) can be used at a time.
`filter_user`	O	ANS	Restricts the log to capture activity for only the Merchant User or Merchant Subuser specified.

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
`bufferid`	N	ID of buffer, if requested with `bufferid=yes`

4.6.4 Retrieve Log Buffer

Retrieve an in-memory log buffer.

This stops and retrieves an in-memory log buffer that was started with a Write log to Buffer request. If you request a specific buffer by passing its buffer ID, then that buffer is returned. Otherwise, this returns the default buffer.

`Access level:` Section 2.4.1: Admin User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**setlogging**
setlogging	Y		**getbuffer**
bufferid	O	N	Buffer ID, as returned from Write log to Buffer request. If not specified, the request will return the default buffer.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
buffer	ANS	In-memory buffer of the log

4.7 System Maintenance

4.7.1 Get Maintenance Level

Get the current maintenance level.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**sysinfo**
sysinfo	Y		**maintenance**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
maintenance	A	Current Maintenance mode. Possible values: • none • strict • limited

4.7.2 Set Maintenance Level

Put the Monetra Server into one of three maintenance modes: STRICT, LIMITED, or NONE. These are typically used when you are upgrading Monetra and need to perform an import/ export on a live system.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**maintenance**
level	Y	A	Maintenance level. Possible values: none - Open up everything for processing (default) limited - Disable add/edit/delete user requests and settlement requests but allows standard transactions strict - Disallow any requests besides Set Maintenance Level, Import Database, and Export Database

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.7.3 Export Database

Export the existing Monetra database to a file. This is normally used when you are upgrading Monetra to a newer version that has a database schema break from the current one.

This is what will be exported during each stage:

- Stage 1 - Merchant accounts
- Stage 2 - CardShield keys/devices
- Stage 3 - Secure stored data (cards on file, recurring billing, etc)
- Stage 4 - Non-historic data (unsettled transactions, etc)
- Stage 5 - Historic data (settled batches, failed transactions, etc)

You can specify a specific stage or a range of stages (1-5 for all) in the stages parameter.

The export key returned in the response parameter key is critically important for the subsequent **import** request. Please secure this key somewhere safe.

Note: The database must be exported to the local file system, and Monetra must have permissions to write to the location.

Access level: Section 2.4.1: Admin User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**export**
file	Y	ANS	Absolute path to file for import/export
stages	O	NS	Stage or range of stages to export, e.g 1 or 2-4
skip_madmin	O	A	Boolean flag indicating whether or not to leave out Admin Users in the export (see Section 2.3: Users for the distinction). Defaults to no, meaning that Admin Users will be included in the export.
hsm_cardshield_export	O	A	Boolean flag indicating whether or not to export keys from HSM rather than keeping them as HSM-protected. Defaults to no.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
key	X, Max 64	Encryption key. Must be used in the subsequent import request.

4.7.4 Import Database

Import the database file obtained from the previous **export** request into the Monetra database. This is normally used when you are upgrading Monetra to a newer version that has a database schema break from the current one.

Note: The database must be imported from the local file system, and Monetra must have permissions to read from the location.

Access level: Section 2.4.1: Admin User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**import**
file	Y	ANS	Absolute path to file for import/export
key	Y	X, Max 64	Unaltered encryption key obtained in the response to the export request

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.7.5 Clear Transaction History

Clear the transaction history of a variety of record types for all Merchant Users.

By default, this will clear the history of voided, failed, and settled transactions for all of Monetra. The purgehist parameter can be passed to change the record types cleared in this request.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**purgehist**
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. If not specified, assume infinite. See Appendix C.8: Date Formats.
purgehist	O	AS	List of record types to purge from the transaction history, separated by pipes (|). Possible values: uncaptured - Uncaptured transactions failed - Failed transactions voided - Voided transactions settled - Settled transactions settlerec - Settlement records

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.8 Automated Admin Task Management

Monetra has a built-in task scheduler known by the common Unix name of "Cron". It is capable of scheduling various tasks--such as history purges, settlements, and token management--on a periodic basis.

Both Admin Users and Merchant Users can schedule tasks with the Cron subsystem, with some tasks allowed for only one user level and others allowed for both. This section strictly applies to Admin User-level tasks. For Merchant User-level tasks, see Section 5.10: Automated Merchant Task Management.

The Cron subsystem will email the results of each task to the specified addresses. For Administrative User tasks, the Cron email system will use the email as specified in the main Monetra configuration file. Monetra must be properly configured to send emails either via a local instance of `sendmail` or a valid `SMTP` server location. For more information on this, see the Secure Install Guide here [https://www.monetra.com/documentation].

`Access level:` Section 2.4.1: Admin User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`cron`**
`cron`	Y	A	The Cron function to perform. See Section 4.8.1: Admin Cron Functions.
`cron_date`	C	ANS	Frequency to run the task. See Section 4.8.2: Admin Cron Date Format.
`cron_task`	C	A	Admin Cron task to schedule. See Section 4.8.3: Admin Cron Tasks.
`cron_data`	C	N	Data specific to a Cron task. See Section 4.8.4: Admin Cron Data.
`cronid`	C	N, Max 20	Unique Cron identifier, as returned from audit. Only used for `remove` and `run_task` functions.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.8.1 Admin Cron Functions

The following functions may be performed through the Cron subsystem:

cron	Description
add	Add the task to the Cron schedule
list	List all scheduled tasks
remove	Remove the task from the Cron schedule
run_task	Run the task immediately. This will not schedule the task. It will only be run one time.

4.8.2 Admin Cron Date Format

The date format is very flexible to allow a wide range of dates. You can specify one or more days of the week to run the task or certain days of the month, but you can only specify one time per task. The format is:
`<time>|<day/date>[;<day/date>[;...]]`

Format	Description
`<time>`	Represented as `HHMM`, assuming a 24hr clock. (e.g. 1430 is 2:30pm)
`<day/date>`	The standard, three-letter abbreviation for the day of the week, or 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
0200|mon;thu	2:00am every Monday and Thursday
1200|1;15	12:00pm on the 1st and 15th of each month
2200|*	10:00pm everyday

4.8.3 Admin Cron Tasks

Admin Users may specify one of the following tasks:

cron_task	Description
purgehist	Purges historical data from the database for all Merchant Users.
purgeexpired	Purges expired cards/accounts within the token system for all Merchant Users. Same as Section 4.10.2: Delete Expired Tokens.
purgerecurhist	Purges the history of recurring and installment payment attempts within the recurring billing and storage system for all Merchant Users. Same as Section 4.10.3: Clear Token History.
purgeuncaptured	Purges all uncaptured transactions for all Merchant Users.
bigbatchsettle	Settles all open batches for the given Big Batch Account ID.
purgebbhist	Purges historical Big Batch data from the database for all Big Batch Users.
account_updater	Runs the Account Updater subsystem.
purge_account_updater	Purges history of account updates for all Merchant Users.

Note: Depending on the Cron task requested, you might receive a different response. For example, `cron=list` will return a comma-separated list of scheduled tasks with the these headers/fields: `cronid,cron_task,cron_date,cron_data,last_ts,next_ts`

4.8.4 Admin Cron Data

Some of the tasks above require additional data to perform their duty. The table below summarizes any necessary information.

Note: The number of days/months must be positive integers.

cron_data	Description
purgehist	Number of days to keep
purgeexpired	Number of months to keep (optional)
purgerecurhist	Number of days to keep
purgeuncaptured	Number of days to keep
bigbatchsettle	Big Batch Account ID
purgebbhist	Number of days to keep
account_updater	N/A
purge_account_updater	Number of days to keep

4.9 License Management

There are three components that can be added on to the main Monetra license: Monetra Admin GUI, Monetra Client GUI, and UniTerm. Each of these components has a number of licenses registered with Monetra. These license counts are maintained by the Monetra Engine and are licensed parameters within the engine itself; they are not related to the number of individual Monetra Engine licenses you have. Each component is licensed separately with its own license count.

The Admin and Client GUI licenses are tied to a machine. When either of those programs is installed, its license is registered in relation to the machine on which it was installed.

The UniTerm licenses are tied to EMV devices. When a device is used through UniTerm for the first time, it will automatically be registered and attached to a license. If there are no more available licenses, then Monetra will check the last-use timestamp of the oldest registered license. If that license hasn't been used within the last 60 days, then it will become available again for re-issue. This eases the process of dealing with devices and licenses.

Note: If a license is deactivated, it will enter a 7-day waiting period ("purgatory") before becoming available again. More information on this can be found here: https://www.monetra.com/faqs/support/registration-failed-license-limit-exceeded.

4.9.1 Get License Counts

Get three pieces of information for each product: number of licenses currently in use, number of licenses in 7-day waiting period, and total number of licenses in the engine.

`Access level:` Section 2.4.1: Admin User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`productlicense`
`productlicense`	Y		`count`

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code indicating success/fail from processor. See Appendix B.1.3: Processor Codes.

Response Parameters		
Key	**Spec**	**Description**
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
admin_count	N	Number of Admin GUI licenses currently in use
admin_hold	N	Number of Admin GUI licenses currently sitting in the 7-day waiting period
admin_limit	N	Number of Admin GUI licenses registered with the engine (i.e. maximum use count)
client_count	N	Number of Client GUI licenses currently in use
client_hold	N	Number of Client GUI licenses currently sitting in the 7-day waiting period
client_limit	N	Number of Client GUI licenses registered with the engine (i.e. maximum use count)
uniterm_count	N	Number of EMV devices currently registered
uniterm_hold	N	Number of device licenses currently sitting in the 7-day waiting period
uniterm_limit	N	Number of devices allowed to be registered (i.e. maximum device count)

4.9.2 List Active Licenses

Get a datablock of licenses currently in use for all three components (Monetra Admin GUI, Monetra Client GUI, and UniTerm).

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**productlicense**
productlicense	Y		**list**
active	O	A	Boolean value indicating whether to show only active licenses or only inactive licenses. Defaults to yes
product	O	A	Licensed product. Possible values: - ADMIN - CLIENT - UNITERM
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. If not specified, assume infinite. See Appendix C.8: Date Formats.
id	O	N, Max 10	Unique license identifier

Report Fields		
Key	**Spec**	**Description**
id	N, Max 10	Unique license identifier
product	A	Licensed product. Possible values: - ADMIN - CLIENT - UNITERM
active	A	Boolean value indicating whether to show only active licenses or only inactive licenses. Defaults to yes
producttype	ANS	Type of product
serialnum	ANS	Product's serial number
last_access_ts	N	Timestamp of last use, in seconds since the Unix epoch
last_access_username	ANS	Admin User and/or Subuser or Merchant User and/or Subuser that last used the license

4.9.3 Deactivate License

Deactivate the specified license.

Note: To reduce fraud, this sends the license to a 7-day waiting period.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**productlicense**
productlicense	Y		**deactivate**
id	Y	N	Unique number identifying the license to be deactivated, as obtained from the List Licenses request.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.10 Token Management

This section deals with the administration and management of tokens and token groups for all Merchant Users.

4.10.1 Manage Token Groups

This subsection concerns the management of token groups. Token groups exist to allow Merchant Users to share tokens with other Merchant Users. This easily enables a business to aggregate the tokens across all their accounts and make those tokens accessible to any Merchant User that needs them.

Generally, an integrator would use this flow when working with token groups:

1. Create a token group.
2. Add a Merchant User to the token group using the `merch_token_group` parameter during an `adduser` or `edituser` request.
3. Create a token as you normally would.
4. Use the token with any Merchant User in that token group.

There are six different actions that can be performed on token groups: `create`, `edit`, `move`, `delete`, `list`, and `list users`.

4.10.1.1 Create Token Group

Create a new token group.

`Access level:`	Section 2.4.1: Admin User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**recurringtokengroup**
recurringtokengroup	Y		**create**
token_group	Y	ANS, Max 32	Unique name for the token group
flags	O	AS	List of token group flags, separated by pipes (|). Possible values: account_updater - Eligible for account updater. See Section 4.13: Account Updater.
custom_customer_fields	O	ANS, Max 32	List of custom field names to allow when adding/editing customers in this token group, separated by commas (,). The field names can have the character set of 'a-z0-9_-'. If present, this list supersedes and invalidates all field names in merch_customer_fields.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.10.1.2 Edit Token Group

Edit the flags and/or custom customer fields associated with the token group.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters

Key	Req	Spec	Description
`action`	Y		**`recurringtokengroup`**
`recurringtokengroup`	Y		**`edit`**
`token_group`	Y	ANS, Max 32	Unique name for the token group
`flags`	O	AS	List of token group flags, separated by pipes (`|`). Possible values: `account_updater` - Eligible for account updater. See Section 4.13: Account Updater.
`custom_customer_fields`	O	ANS, Max 32	List of custom field names to allow when adding/editing customers in this token group, separated by commas (`,`). The field names can have the character set of '`a-z0-9_-`'. If present, this list supersedes and invalidates all field names in `merch_customer_fields`.

Response Parameters

Key	Spec	Description
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.10.1.3 Move Token Group

Move all the tokens in one token group over to a different token group. This will move every token; there is currently no way to move only select tokens.

Note: After this request, new tokens will continue to be added to the original token group. To stop using the original token group and instead add tokens to the new token group, all Merchant Users assigned to the original token group will need to be edited to use the new token group.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**recurringtokengroup**
recurringtokengroup	Y		**move**
token_group	Y	ANS, Max 32	Unique name for the token group
assign_group	Y	ANS	Unique name for the token group to which all tokens will be moved

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.10.1.4 Delete Token Group

Delete a token group. The token group must not have any tokens, nor can a Merchant User or customer belong to the group.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**recurringtokengroup**
recurringtokengroup	Y		**delete**
token_group	Y	ANS, Max 32	Unique name for the token group

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.10.1.5 List Token Group

Get a datablock of information for either all token groups, a specified token group, or the token group that contains a specified token.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**recurringtokengroup**
recurringtokengroup	Y		**list**
token	O	N	Token, as returned in the Recurring Add request
token_group	O	ANS, Max 32	Unique name for the token group

Report Fields		
Key	**Spec**	**Description**
group	ANS, Max 32	Unique name for the token group
flags	AS	List of token group flags, separated by pipes (|). Possible values: account_updater - Eligible for account updater. See Section 4.13: Account Updater.
custom_customer_fields	ANS, Max 32	List of custom field names to allow when adding/editing customers in this token group, separated by commas (,). The field names can have the character set of 'a-z0-9_-'. If present, this list supersedes and invalidates all field names in merch_customer_fields.
num_users	N	Number of Merchant Users in token group
num_tokens	N	Number of tokens in token group
num_customers	N	Number of customers in token group

4.10.1.6 List Token Group Users

Get a datablock of users that belong to a token group as well as the token group they belong to, optionally limited to a specified token group.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**recurringtokengroup**
recurringtokengroup	Y		**listusers**
token_group	O	ANS, Max 32	Unique name for the token group

Report Fields		
Key	**Spec**	**Description**
user	ANS	Name of Merchant User
group	ANS, Max 32	Unique name for the token group

4.10.2 Delete Expired Tokens

Remove all tokens for all Merchant Users from the Card Storage subsystem that have accounts with expired cards. This will only act on store and nontrans tokens. This can be automated on a periodic basis with Monetra's Cron subsystem using cron_task=purgeexpired. To instead update the account information for expired tokens, see Section 4.13: Account Updater.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**recurringpurgeexpired**
keep_months	O	N	Number of months back to keep

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.10.3 Clear Token History

Remove the token history of transactions that were run from the Recurring Billing subsystem for all Merchant Users. This affects only transactions that were scheduled and used with a recurring or installment token. To schedule this request on a periodic basis with Monetra's Cron subsystem, use cron_task=purgerecurhist.

This does not clear the transaction history; it clears the history of token usage, as viewed with the List Recurring History request. To clear the transaction history, see Section 4.7.5: Clear Transaction History.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**recurringpurgehist**
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. If not specified, assume infinite. See Appendix C.8: Date Formats.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11 Big Batch Aggregation

This section deals with the administration and management of Big Batch settlement through the Big Batch Aggregator subsystem, which enables an integrator to aggregate batches from multiple Merchant Users and settle them as one unit. For details on adding this feature to an existing Monetra installation, contact support@monetra.com.

The basic concept behind Big Batch aggregation is that one administrative user will combine separate batches from multiple merchants into one Big Batch file, which will be sent to the processor as a single unit. To achieve this, a Big Batch Admin User must first be set up. This user will then have multiple Big Batch Merchant Users added to it. Each Big Batch Merchant User on that account will correspond to a Merchant User that will be settling into this aggregated batch. When a Merchant User settles their batch, it will go through strict validation checks and be added to the Big Batch. When all the Merchant User batches have been added to the Big Batch, the Big Batch Admin User will then settle it, which will send the Big Batch online to the chosen processor. You can determine which processors support Big Batch settlement through Monetra by sending a List Processors request and checking the `features` column for the keyword `BIGBATCH`.

The process works like this:

1. Create a Big Batch Admin User.
2. Create a Big Batch Merchant User for each Merchant User that will settle into this Big Batch Admin User.
3. Link each Merchant User to its associated Big Batch Merchant User in the settlement route.
4. When desired, settle the Merchant User's open batch. This will actually add the batch to the Big Batch Admin User's open batch.
5. When desired, settle the Big Batch Admin User's batch. This will settle all the individual Merchant User batches as one aggregated batch.

4.11.1 Manage Big Batch Admin Users

4.11.1.1 Add Big Batch Admin User

Add a new Big Batch Admin User.

`Access level:`	Section 2.4.1: Admin User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**addacct**
descr	Y	ANS	User-provided description
proc	Y	A	Processing institution
email	O	ANS	Email address for sending notices/cron emails
In addition to the fields above, you must also populate processor-specific fields, which can be determined with a procfields request. Only the fields designated as "submitter only" are relevant.			

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
bbacctid	N	Unique ID generated for each Big Batch Admin User

4.11.1.2 Edit Big Batch Admin User

Edit an existing Big Batch Admin User.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**editacct**
bbacctid	Y	N	Unique ID generated for each Big Batch Admin User
descr	O	ANS	User-provided description
email	O	ANS	Email address for sending notices/cron emails
In addition to the fields above, you may also populate processor-specific fields, which can be determined with a procfields request. Only the fields designated as "submitter only" are relevant.			

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11.1.3 Delete Big Batch Admin User

Delete an existing Big Batch Admin User.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**delacct**
bbacctid	Y	N	Unique ID generated for each Big Batch Admin User

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11.1.4 List Big Batch Admin Users

Get a datablock of Big Batch Admin Users.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**listacct**
bbacctid	O	N	Unique ID generated for each Big Batch Admin User

Report Fields		
Key	**Spec**	**Description**
bbacctid	N	Unique ID generated for each Big Batch Admin User
proc	A	Processing institution
descr	ANS	User-provided description
num_merchants	N	Number of Big Batch Merchant Users

4.11.1.5 Get Big Batch Admin User Details

Get the information associated with the provided Big Batch Admin User.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**getacct**
bbacctid	Y	N	Unique ID generated for each Big Batch Admin User

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
descr	ANS	User-provided description
proc	A	Processing institution
email	ANS	Email address for sending notices/cron emails
In addition to the fields above, any populated processor-specific fields will also be returned.		

4.11.2 Manage Big Batch Merchant Users

4.11.2.1 Add Big Batch Merchant User

Add a new Big Batch Merchant User. The bbmerchid and authtoken returned in the response will be used to link the settle route of the Merchant User to the newly created Big Batch Merchant User.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**addmerch**
bbacctid	Y	N	Unique ID generated for each Big Batch Admin User
descr	Y	ANS	User-provided description
indcode	Y	A, Max 2	Industry code for processing transactions. See Appendix C.5: Industry Codes.
cardtypes	Y	AS	List of card types, separated by pipes (|) or pluses (+). See Appendix C.1: Card Types.
merch_tz	O	ANS	Merchant's time zone. See Section 4.5.4: List Time Zones.
In addition to the fields above, you must also populate processor-specific fields, which can be determined with a procfields request. Only the fields NOT designated as "submitter only" are relevant.			

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
bbmerchid	N	Unique ID generated for each Big Batch Merchant User
authtoken	AN	Unique password generated for each Big Batch Merchant User

4.11.2.2 Edit Big Batch Merchant User

Edit an existing Big Batch Merchant User.

Access level: Section 2.4.1: Admin User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**editmerch**
bbmerchid	Y	N	Unique ID generated for each Big Batch Merchant User
descr	O	ANS	User-provided description
indcode	O	A, Max 2	Industry code for processing transactions. See Appendix C.5: Industry Codes.
cardtypes	O	AS	List of card types, separated by pipes (|) or pluses (+). See Appendix C.1: Card Types.
merch_tz	O	ANS	Merchant's time zone. See Section 4.5.4: List Time Zones.
In addition to the fields above, you may also populate processor-specific fields, which can be determined with a procfields request. Only the fields NOT designated as "submitter only" are relevant.			

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11.2.3 Delete Big Batch Merchant User

Delete an existing Big Batch Merchant User.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**delmerch**
bbmerchid	Y	N	Unique ID generated for each Big Batch Merchant User

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11.2.4 List Big Batch Merchant Users

Get a datablock of Big Batch Merchant Users for the provided Big Batch Admin User.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**listmerch**
bbacctid	Y	N	Unique ID generated for each Big Batch Admin User

Report Fields		
Key	**Spec**	**Description**
bbacctid	N	Unique ID generated for each Big Batch Admin User
bbmerchid	N	Unique ID generated for each Big Batch Merchant User
descr	ANS	User-provided description

4.11.2.5 Get Big Batch Merchant User Details

Get the information associated with the provided Big Batch Merchant User.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchmng**
bigbatchmng	Y		**getmerch**
bbmerchid	Y	N	Unique ID generated for each Big Batch Merchant User

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
bbacctid	N	Unique ID generated for each Big Batch Admin User
bbmerchid	N	Unique ID generated for each Big Batch Merchant User
authtoken	AN	Unique password generated for each Big Batch Merchant User
descr	ANS	User-provided description
proc	A	Processing institution
indcode	A, Max 2	Industry code for processing transactions. See Appendix C.5: Industry Codes.
cardtypes	AS	List of card types, separated by pipes (|) or pluses (+). See Appendix C.1: Card Types.
zipcode	AN	The zip code for the retail location of the Merchant or the Merchant's headquarters. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.

Response Parameters		
Key	**Spec**	**Description**
merch_tz	ANS	Merchant's time zone. See Section 4.5.4: List Time Zones.
In addition to the fields above, any populated processor-specific fields will also be returned.		

4.11.3 Manage Big Batches

This section deals with the management of batches in the Big Batch subsystem.

4.11.3.1 Settle Big Batch

Send an aggregated Big Batch online for settlement with the processor.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatchsettle**
bbbatchid	Y	N	Unique ID generated for each Big Batch
resubmit	O	A	Boolean flag indicating whether or not the batch should be resubmitted for settlement. Only valid for batches that have been previously settled or force-settled. Defaults to no.
rfr	O	A	Boolean flag indicating whether or not to request the settlement status of this batch in the processor's system instead of attempting to settle it. Defaults to no.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11.3.2 List Big Batches

Get a datablock of Big Batches in the system, either for the entire system, for a specific Big Batch Admin User, for a specific Big Batch, or based on the state of a Big Batch.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatch**
bigbatch	Y		**list**
bbacctid	O	N	Unique ID generated for each Big Batch Admin User
bbbatchid	O	N	Unique ID generated for each Big Batch
state	O	AS	List of batch statuses, separated by pipes (|). Possible values: open - Batch can be added to. closed - Batch cannot be added to, but settlement has not occurred yet. pending - Settlement is in progress, and the batch is locked. attempted - Batch upload was attempted, but there was a communication error. Must be manually resubmitted for settlement. settled - Batch was settled successfully. forcesettled - Batch was manually settled offline.

Report Fields		
Key	**Spec**	**Description**
bbacctid	N	Unique ID generated for each Big Batch Admin User
bbbatchid	N	Unique ID generated for each Big Batch
state	AS	List of batch statuses, separated by pipes (|). Possible values: open - Batch can be added to. closed - Batch cannot be added to, but settlement has not occurred yet. pending - Settlement is in progress, and the batch is locked.

Report Fields		
Key	**Spec**	**Description**
		attempted - Batch upload was attempted, but there was a communication error. Must be manually resubmitted for settlement. settled - Batch was settled successfully. forcesettled - Batch was manually settled offline.
settlets	N	Timestamp of last access or use, in seconds since the Unix epoch
num_trans	N	Total number of authorizations and returns
amount_trans	M	Aggregate authorization amounts minus aggregate return amounts
num_level3	N	Number of transactions with Level 3 line items
num_merchant_batches	N	Number of batches aggregated into this Big Batch

4.11.3.3 List Big Batch Merchant User Batches

Get a datablock of individual merchant batches (owned by a Big Batch Merchant User) that exist within an aggregate Big Batch (owned by a Big Batch Admin User). The batches can be filtered by the provided Big Batch Merchant User, a specific Big Batch, or the state of a Big Batch.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`bigbatch`**
`bigbatch`	Y		**`listmerchbatches`**
`bbmerchid`	C	N	Unique ID generated for each Big Batch Merchant User
`bbbatchid`	C	N	Unique ID generated for each Big Batch
`state`	C	AS	List of batch statuses, separated by pipes (|). Possible values: `open` - Batch can be added to. `closed` - Batch cannot be added to, but settlement has not occurred yet. `pending` - Settlement is in progress, and the batch is locked. `attempted` - Batch upload was attempted, but there was a communication error. Must be manually resubmitted for settlement. `settled` - Batch was settled successfully. `forcesettled` - Batch was manually settled offline.

Report Fields		
Key	**Spec**	**Description**
`bbmerchid`	N	Unique ID generated for each Big Batch Merchant User
`bbbatchid`	N	Unique ID generated for each Big Batch
`batchnum`	N, Max 10	Batch number
`submit_ts`	N	Timestamp of when batch was added to Big Batch, in seconds since the Unix epoch
`state`	AS	List of batch statuses, separated by pipes (|). Possible values: `open` - Batch can be added to. `closed` - Batch cannot be added to, but settlement has not occurred yet. `pending` - Settlement is in progress, and the batch is locked. `attempted` - Batch upload was attempted, but there was a communication error. Must be manually resubmitted for settlement. `settled` - Batch was settled successfully.

Report Fields		
Key	**Spec**	**Description**
		forcesettled - Batch was manually settled offline.
settlets	N	Timestamp of last access or use, in seconds since the Unix epoch
num_trans	N	Total number of authorizations and returns
amount_trans	M	Aggregate authorization amounts minus aggregate return amounts

4.11.3.4 List Failed Transactions

Get a datablock of failed transactions in settled batches.

Transactions that fail during settlement have different effects on the batch depending on whether the batch was transmitted with the online protocol or the Big Batch protocol. When a batch is settled using the online protocol, a failed transaction will cause the entire batch to be rejected. The merchant will then need to determine the cause of the failure (possibly by working with the processor) and correct the issue before attempting the settlement again. No money will move until the entire batch is correct and successfully submitted.

Big Batch settlement is different. When a batch is settled using the Big Batch protocol, the batch will be approved or rejected based on considerations other than transactions. If everything is correct with the submission, then the batch will be approved. At this point, if an individual transaction fails, then it will not affect anything else in the batch. That is, money will move for the transactions that were settled successfully, but no money will move for the failed transactions. If there are failed transactions in a Big Batch, then a merchant will need to determine the cause of the failure (possibly by working with the processor) and correct the issue, and then resettle those transactions.

This request can be used to determine which of the transactions in the settled batch failed. This request is only relevant for processors that return a settlement response indicating the status of individual transactions.

Access level: Section 2.4.1: Admin User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatch**
bigbatch	Y		**listfailedtrans**
bbacctid	O	N	Unique ID generated for each Big Batch Admin User
bbmerchid	C	N	Unique ID generated for each Big Batch Merchant User
bbbatchid	O	N	Unique ID generated for each Big Batch
batchnum	O	N, Max 10	Batch number

Report Fields		
Key	**Spec**	**Description**
bbacctid	N	Unique ID generated for each Big Batch Admin User
bbmerchid	N	Unique ID generated for each Big Batch Merchant User
bbbatchid	N	Unique ID generated for each Big Batch
batchnum	N, Max 10	Batch number
submit_ts	N	Timestamp of when batch was added to Big Batch, in seconds since the Unix epoch
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
trantype	A	Transaction type. Possible transaction types are any listed in Chapter 6: *Merchant Subuser Actions*.
cardtype	A	Card type detected. See Appendix C.1: Card Types.
amount	M	Amount of transaction

4.11.3.5 Clear Settled

Remove settled and force-settled transactions from the database.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatch**
bigbatch	Y		**purge**
bdate	Y	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	Y	ANS	Ending date. If not specified, assume infinite. See Appendix C.8: Date Formats.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11.3.6 Purge Merchant Batch

Remove the provided Big Batch Merchant User's batch from the aggregate big batch. This does not delete any transactions or batches at the Merchant User level.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatch**
bigbatch	Y		**purgemerchbatch**
bbmerchid	Y	N	Unique ID generated for each Big Batch Merchant User
bbbatchid	Y	N	Unique ID generated for each Big Batch
batch	Y	N, Max 10	Batch number

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.11.3.7 Force Settle Big Batch

Mark a Big Batch as settled within Monetra *without* sending it online to the processor or requesting funding. See Section 5.5.4: Force-Settle Batch for a similar action at the Merchant User-level.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**bigbatch**
bigbatch	Y		**forcesettle**
bbbatchid	Y	N	Unique ID generated for each Big Batch

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.12 Push Notification Management

With Push Notifications/WebHooks, it is possible to send notifications via HTTPS-POST requests to a merchant-defined endpoint for the occurrence of merchant-level events. There are a series of flags that can be configured to determine which events will trigger a notification. Different merchants can be assigned to different endpoints.

To limit load on the system, notifications are aggregated and sent to each endpoint in 30-second batches.

If Monetra sends a push notification to an endpoint and is not able to connect or receives an HTTP response code that is not 200, then the POST request will reattempted later.

4.12.1 Notification Format

The data sent in the notification is a JSON object named `transactions`, which is a JSON array of transactions. All transactions have a similar structure, with only a small number of fields differing between transaction types. Below is a table detailing which fields are received for which transaction types and what the various fields mean.

Key	Main Merchant User Actions	Merchant Subuser Actions	Settlement	Spec	Description
`username`	Y	Y	Y	ANS	Merchant User or Merchant Subuser that ran the transaction
`action`	Y	Y	Y	AS	Action performed. Valid values are `admin` or any of the actions in Chapter 6: *Merchant Subuser Actions*.
`admin`	Y			AS	Admin action performed. Valid values are any of the admin actions in Chapter 5: *Main Merchant User Actions*.
`code`	Y	Y	Y	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`cardtype`	C	Y	Y	AS	For settlement or force settlement, list of supported card types, separated by pipes (`|`). Otherwise, card type used to run transaction. See Appendix C.1: Card Types.
`ttid`	Y	Y	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
`ts`	Y	Y	Y	N	Timestamp of when transaction was run, in seconds since the Unix epoch

Key	Main Merchant User Actions	Merchant Subuser Actions	Settlement	Spec	Description
tranflags		Y		ANS	Monetra-defined transaction flags. See Appendix B.5.18: Transaction Flags.
entrymode		Y		A, Fixed 1	Monetra-defined value indicating how the transaction data was entered. See Appendix B.4: Alphabetical Listing.
merchant	Y	Y	Y	ANS	JSON object containing two children: indcode - Industry code proc - Processing institution
request	Y	Y	Y	ANS	JSON object containing various request parameters sent with the original transaction request, excluding sensitive data. See Appendix A: *Request Parameters*.
response	Y	Y	Y	ANS	JSON object containing various response parameters returned with the original transaction response. See Appendix B: *Response Parameters*.
interchange	Y	Y	Y	ANS	If enabled, JSON object containing various intercharge data returned with the original transaction response. See Appendix B.5: Raw Response Codes.

4.12.2 Add Endpoint

Add a new endpoint.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**pushnotification**
pushnotification	Y		**add**
display_name	Y	ANS	Human-readable display name
url	Y	ANS	URL of the endpoint, prefixed with https://
eventflags	Y	AS	List of push notification flags, separated by pipes (|). Possible values:

Request Parameters			
Key	Req	Spec	Description
			Transaction Types: auth - Include normal auths (sale, preauth, force, return) verify - Include verification requests (balanceinq, avsonly) settle - Include batch settlement requests edit - Include edit requests (adjust, preauthcomplete, fieldedit) void - Include void/reversal requests Modifiers: declined - Include declined transactions interchange - Include interchange data Card Types: credit - Include credit card transactions debit - Include debit card transactions gift - Include private label gift transactions ach - Include ACH transactions ebt - Include EBT transactions
authtype	Y	A	Type of HTTP authentication to use. Possible values: none basic
authname	O		Currently unused. In the future, this will be the HTTP header name.
authdata	C	ANS	For authtype=basic authentication, the base64-encoded user:password

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
id	N	Unique ID generated for each Push Notification endpoint

4.12.3 Edit Endpoint

Edit an existing endpoint.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**pushnotification**
pushnotification	Y		**edit**
id	Y	N	Unique ID generated for each Push Notification endpoint
display_name	O	ANS	Human-readable display name
url	O	ANS	URL of the endpoint, prefixed with https://
eventflags	O	AS	List of push notification flags, separated by pipes (|). Possible values: Transaction Types: auth - Include normal auths (sale, preauth, force, return) verify - Include verification requests (balanceinq, avsonly) settle - Include batch settlement requests edit - Include edit requests (adjust, preauthcomplete, fieldedit) void - Include void/reversal requests

Request Parameters			
Key	**Req**	**Spec**	**Description**
			Modifiers: declined - Include declined transactions interchange - Include interchange data Card Types: credit - Include credit card transactions debit - Include debit card transactions gift - Include private label gift transactions ach - Include ACH transactions ebt - Include EBT transactions
authtype	O	A	Type of HTTP authentication to use. Possible values: none basic
authname	O		Currently unused. In the future, this will be the HTTP header name.
authdata	C	ANS	For authtype=basic authentication, the base64-encoded user:password

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.12.4 Delete Endpoint

Delete an existing endpoint.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**pushnotification**
pushnotification	Y		**del**
id	Y	N	Unique ID generated for each Push Notification endpoint

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to Monetra-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code indicating success/ fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

4.12.5 List Endpoints

Get a datablock of configured endpoints, optionally filtered by endpoint id.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**pushnotification**
pushnotification	Y		**list**
id	O	N	Unique ID generated for each Push Notification endpoint

Report Fields		
Key	**Spec**	**Description**
id	N	Unique ID generated for each Push Notification endpoint
display_name	ANS	Human-readable display name
url	ANS	URL of the endpoint, prefixed with https://
eventflags	AS	List of push notification flags, separated by pipes (|). Possible values:

Report Fields		
Key	**Spec**	**Description**
		Transaction Types: auth - Include normal auths (sale, preauth, force, return) verify - Include verification requests (balanceinq, avsonly) settle - Include batch settlement requests edit - Include edit requests (adjust, preauthcomplete, fieldedit) void - Include void/reversal requests Modifiers: declined - Include declined transactions interchange - Include interchange data Card Types: credit - Include credit card transactions debit - Include debit card transactions gift - Include private label gift transactions ach - Include ACH transactions ebt - Include EBT transactions
authtype	A	Type of HTTP authentication to use. Possible values: none basic

4.12.6 List Users

Get a datablock of Merchant Users registered to an endpoint, optionally filtered by endpoint id. A Merchant User can be registered to an endpoint during an Add Merchant User or Edit Merchant User request using the merch_pushnotification_id parameter.

Access level: Section 2.4.1: Admin User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**pushnotification**
pushnotification	Y		**listusers**
id	O	N	Unique ID generated for each Push Notification endpoint

Report Fields		
Key	**Spec**	**Description**
user	ANS	Name of user
display_name	ANS	Human-readable display name
endpoint_id	N	Unique ID generated for each Push Notification endpoint
endpoint_url	ANS	URL of the endpoint, prefixed with https://

4.13 Account Updater

The Account Updater subsystem allows merchants to retrieve updated card numbers and expiration dates for customer accounts (whether due to a card being lost or stolen, or simply a new card being issued).

Currently, the Account Updater subsystem is only supported through TSYS. Enrolling in the program requires a special setup from TSYS and a sponsoring bank for approval. Note that only the account updates are processed through TSYS; no other processing (like authorization or settlement) with TSYS is required.

4.13.1 Merchant User's Tokens

When a Merchant User is created or edited, the parameter `merch_flags` can be passed with a value of `account_updater` to mark the Merchant User's tokens as eligible for updates. See Section 4.3.1: Add Merchant User.

Note: `merch_flags` cannot be used if the Merchant User is part of a token group. See Section 4.13.2: Token Groups.

4.13.2 Token Groups

When a token group is created or edited, the parameter `flags` can be passed with a value of `account_updater` to mark the group's tokens as eligible for updates. See Section 4.10.1.1: Create Token Group.

4.13.3 Automated Account Updates

The Cron subsystem can be used to automate account updates. For details on this, see Section 4.8: Automated Admin Task Management and Section 4.8.3: Admin Cron Tasks, specifically the tasks `account_updater` and `purge_account_updater`.

4.13.4 List Counts

Get a datablock of update counts for each Merchant User and token group, optionally filtered by Merchant User or token group.

`Access level:`	Section 2.4.1: Admin User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**liststats**
liststats	Y		**accountupdater**
bdate	Y	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	Y	ANS	Ending date. If not specified, assume infinite. See Appendix C.8: Date Formats.
user	O	ANS	Name of user. If not sent, applies to all users.
token_group	O	ANS, Max 32	Unique name for the token group

Report Fields		
Key	**Spec**	**Description**
user	ANS	Name of user
token_group	ANS, Max 32	Unique name for the token group
update_count	N	Number of tokens updated

5 Main Merchant User Actions

The following sections in this chapter detail the actions that main Merchant Users with sufficient privileges are allowed to perform.

Please review the required permission level to send main Merchant User actions. See Section 2.4.2: Merchant User Authentication.

5.1 Merchant Subuser Management

Subusers allow multiple people to run transactions with varying permission levels. If more than one person will have access to the system, then it is recommended to create subusers that are granted the minimum permissions they require to perform their duties. See Section 2.3.3: Permissions for information on assigning permissions.

When sending transactions as a subuser, this format is used:

```
merchuser:subuser
```

For example, if Merchant User `Store_Manager` created the subuser `clerk1`, then that subuser would use `username=Store_Manager:clerk1` to run transactions through Monetra.

The actions outlined in this section provide a means to manage all subusers on the account.

Note: There is a major difference between "subusers" and "subaccounts"; the two terms should not be confused. Subusers deal with permissions and running transactions, while subaccounts deal with processing and splitting routes.

5.1.1 Add Merchant Subuser

Add a new subuser.

Note: There is no limit to the number of subusers that can be added for a particular account. We recommend you add one for each individual that will have access to the system, especially for reports and/or transactions that return funds.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`admin`**
`admin`	Y		**`subuseradd`**
`user`	Y	ANS, Max 256	Username
`pwd`	Y	ANS, Max 256	Password. Must be mix of uppercase letters, lowercase letters, numbers, and special characters.
`password_expired`	O	A	Boolean flag indicating whether or not to set the user's password as expired, which will force a password change at next login. Defaults to `no`.

Request Parameters			
Key	**Req**	**Spec**	**Description**
admintypes	Y	AS	List of administrative permissions for the subuser, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
trantypes	Y	A	List of permitted transaction types, separated by pipes (|), or all for all transaction types. Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
userflags	O	AS	List of flags for the subuser, separated by pipes (|). Possible values: obscure - Always obscure sensitive data unattended - Prevent the password from expiring (useful for connected integrations) selfviewonly - Restrict reports to show only this subuser's data authas - Allow subuser to run transactions as other subusers

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.1.2 Edit Merchant Subuser

Edit an existing subuser.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`admin`**
`admin`	Y		**`subuseredit`**
`user`	Y	ANS, Max 256	Username
`pwd`	O	ANS, Max 256	Password. Must be mix of uppercase letters, lowercase letters, numbers, and special characters.
`password_expired`	O	A	Boolean flag indicating whether or not to set the user's password as expired, which will force a password change at next login. Defaults to `no`.
`admintypes`	O	AS	List of administrative permissions for the subuser, separated by pipes (|), or `all` for all permissions. Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
`trantypes`	O	A	List of permitted transaction types, separated by pipes (|), or `all` for all transaction types. Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
`userflags`	O	AS	List of flags for the subuser, separated by pipes (|). Possible values: `obscure` - Always obscure sensitive data `unattended` - Prevent the password from expiring (useful for connected integrations) `selfviewonly` - Restrict reports to show only this subuser's data `authas` - Allow subuser to run transactions as other subusers

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.1.3 Delete Merchant Subuser

Delete an existing subuser from the system.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**subuserdel**
user	Y	ANS, Max 256	Username

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.1.4 List Merchant Subusers

Get a datablock of subusers (and associated data) assigned to the Merchant User, optionally limited to a single provided subuser.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**subuserlist**
user	O	ANS, Max 256	Username

Report Fields		
Key	**Spec**	**Description**
user	ANS, Max 256	Username
master	A	Boolean flag indicating whether or not the account is a master account
trantypes	A	List of permitted transaction types, separated by pipes (|), or all for all transaction types. Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
admintypes	AS	List of administrative permissions for the subuser, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
userflags	AS	List of flags for the subuser, separated by pipes (|). Possible values: obscure - Always obscure sensitive data unattended - Prevent the password from expiring (useful for connected integrations) selfviewonly - Restrict reports to show only this subuser's data authas - Allow subuser to run transactions as other subusers
obscure	A	Boolean flag indicating whether or not to always obscure sensitive data

Report Fields		
Key	**Spec**	**Description**
unattended	A	Boolean flag indicating whether or not the account is marked as unattended
pass_expire_secs	N, Max 19	Number of seconds until the password expires (or -1 for never or 0 for expired)
locked	A	Boolean flag indicating whether or not the account is locked

5.1.5 Get Permissions

Get a datablock of administrative actions and transaction types allowed for the calling Merchant User/Subuser.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**getperms**

Report Fields		
Key	**Spec**	**Description**
trantypes	A	List of permitted transaction types, separated by pipes (|), or all for all transaction types. Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
admintypes	AS	List of administrative permissions for the subuser, separated by pipes (|), or all for all permissions. Valid values are any of the actions in Chapter 5: *Main Merchant User Actions*.
userflags	AS	List of flags for the subuser, separated by pipes (|). Possible values: obscure - Always obscure sensitive data unattended - Prevent the password from expiring (useful for connected integrations) selfviewonly - Restrict reports to show only this subuser's data authas - Allow subuser to run transactions as other subusers

Report Fields		
Key	**Spec**	**Description**
obscure	A	Boolean flag indicating whether or not to always obscure sensitive data
unattended	A	Boolean flag indicating whether or not the account is marked as unattended

5.1.6 Change Password

Change the password of the calling user/subuser.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**chngpwd**
pwd	Y	ANS, Max 256	Password. Must be mix of uppercase letters, lowercase letters, numbers, and special characters.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.1.7 Unlock Account

Unlock a Merchant Subuser under the calling Merchant User. A Merchant User can unlock only the subusers that it has created.

An account can be locked due to exceeding the limit of password tries. See the entry for password_lockout_seconds in the Secure Install Guide [https://www.monetra.com/documentation] for more information on password lockouts.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**unlockacct**
user	Y	ANS, Max 256	Subuser to unlock

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.1.8 Get Merchant Information

Get the merchant parameters (most start with merch_*) that were set when creating or editing this merchant.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**merchinfo**
merchinfo	Y		**termparams**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.

Response Parameters		
Key	**Spec**	**Description**
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
countrycode	N, 3	A reference code to the country the Merchant is processing in. (ISO 3166-1 numeric, ex: USA = 840).
currencycode	N, 3	A reference code to the currency the Merchant is processing in. (ISO 4217, ex: USA = 840).
merch_name	ANS	Merchant's name
merch_addr1	ANS	Merchant's address - line 1
merch_addr2	ANS	Merchant's address - line 2
merch_addr3	ANS	Merchant's address - line 3
merch_phone	ANS	Merchant's phone number
merch_email	ANS	Merchant's email
merch_url	ANS	Merchant's website URL
merch_lang	A	2-character language code. Possible values: en for English fr for French es for Spanish de for German it for Italian
merch_cashback	ANS	List of dollar amounts to display on screen during cash back prompting, separated by pipes (|). All values are whole dollar amounts, and the letter O can be used to indicate the ability to enter a custom amount. For example, 5|10|25|O would prompt for $5, $10, $25, or Other.
merch_cashbackmax	M	Maximum cash back amount, if allowed to entered a custom value
merch_cashback_purchmin	M	Minimum cash back purchase amount
merch_tippercent	M	Merchant tip percentages to display
merch_msr_nosig_limit	M	Limit to not require a signature for MSR

Response Parameters		
Key	**Spec**	**Description**
`merch_enc_provider`	A	Processor-level, end-to-end encryption. Currently, only a value of `bluefin` is supported
`merch_l3_commoditycode`	AN	Level III Item Commodity Code. International description code of the individual good or service being supplied. Required if `prepop_level3` is set in `merch_flags`.
`merch_l3_description`	ANS	Item Descriptor. Merchant-defined description of the item or service being sold. Required if `prepop_level3` is set in `merch_flags`.
`merch_l3_productcode`	ANS	Product Code. Merchant-defined code for the item being purchased, such as a UPC number, SKU, or Item Number. Required if `prepop_level3` is set in `merch_flags`.
`merch_tax_rate`	M	Tax rate as a percentage between 0.1 and 49.99. Required if `prepop_level3` is set in `merch_flags`.
`merch_token_group`	ANS, Max 32	Token group. See Section 4.10.1: Manage Token Groups.
`merch_custom_fields`	ANS	List of custom field names to allow for specifying custom fields sent with a transaction, separated by commas (`,`). The field names can have the character set of `a-z0-9_-` and be up to 32 characters each. * During each transaction request (e.g. **`sale`**, **`preauth`**, **`return`**), one or more of the configured merchant custom fields may be specified using the standard key/value pair format as defined by the protocol. This data will be stored with the transaction record. * Merchant custom fields can be updated during `fieldedit`, **`adjust`**, **`preauthcomplete`**, or **`capture`** requests. * For each merchant custom field configured for a merchant, a new column will appear in the `gut`, `gl`, and `gft` reports and will contain any custom data presented with each transaction..
`merch_req_fields`	AS	List of field names to require for every transaction, separated by commas (`,`). Possible values:

Response Parameters		
Key	**Spec**	**Description**
		tax examount custref ordernum stationid clerkid comments cardholdername descmerch descloc cvv2 (but only required if online and not EMV or Swipe) zip (but only required if online and not EMV or Swipe) any of the merch_custom_fields

5.1.9 Get Merchant Configuration

Get a datablock of configurations for each card type (Visa, ACH, etc.) that the calling Merchant User supports, as set with Section 4.3.1: Add Merchant User or Section 4.3.2: Edit Merchant User. This can be used to determine the route that Monetra will take when presented with each cardtype.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**merchinfo**

Report Fields		
Key	**Spec**	**Description**
cardtype	A	Card type detected. See Appendix C.1: Card Types.
auth_proc	ANS	Processor configured for authorization route
settle_proc	ANS	Processor configured for settlement route
auth_merchid	ANS	MID or TID configured for authorization route (if applicable)
settle_merchid	ANS	MID or TID configured for settlement route (if applicable)

Report Fields		
Key	**Spec**	**Description**
auth_sub	N, Max 10	Authorization subaccount
settle_sub	N, Max 10	Settlement subaccount
indcode	A, Max 2	Industry code for processing transactions. See Appendix C.5: Industry Codes.
trantypes	A	List of permitted transaction types, separated by pipes (|), or all for all transaction types. Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
emv	AS	List of EMV entry modes, separated by pipes (|). Possible values: CONTACT CONTACTLESS
mac_required	AS	Type of MAC'ing required, if any. Possible values: none yes dukpt
gift_binrange	NS	BIN range (gift cards only). See Appendix C.10: BIN Range Format.
can_standin	A	Boolean flag indicating whether or not stand-in (store and forward) is allowed
auth_procfeatures	ANS	List of processor features for the authorization route, separated by pipes (|). See Appendix C.4: Processor Features.
settle_procfeatures	ANS	List of processor features for the settlement route, separated by pipes (|). See Appendix C.4: Processor Features.

5.2 Tokens and Recurring Payments

This section covers two separate but intertwined subsystems: Card Storage and Recurring Billing. The Card Storage subsystem in Monetra provides an advanced and security-compliant method that uses military-grade encryption and modern security practices for storing and replacing sensitive account data with sanitized, pseudo card numbers (tokens) that can be used for future transactions. The Recurring Billing subsystem in Monetra offers merchants a robust, secure, and completely integrated way to establish and automatically process repetitive payment transactions, such as monthly subscriptions.

The ability to securely store sensitive account data and automatically perform recurring billing operations adds convenience for day-to-day business processes while maintaining full compliance with credit card industry security guidelines.

Card data stored in Monetra is replaced in the POS system by a token, eliminating the sensitive information that otherwise could be subject to compromise. The token is then used in place of the actual card number when performing transactions (such as sales) for that card. When a transaction using that token is performed, Monetra will internally replace the token with the corresponding card's data, so the actual card data is never exposed.

There are four different types of tokens:

`store` - This will keep the card in the system for later use at an unspecified time. This could be used by an Ecommerce site storing a card with a user's profile.

`recurring` - This will run a sale at a set interval for a specified duration or indefinitely. A magazine could use this to bill its customers every month.

`installment` - This will run a sale at a set interval until the total amount is paid off. This is relevant for a merchant providing layaway plans to customers.

`nontrans` - This token cannot be used for running transactions; the token is only used to store the card information in a safe and secure way. One use case with this type is a parking garage that needs to track time in and time out based on the card presented.

> Note: All recurring/installment transactions run at the same time each day (for the transactions that are set to run that day). The time is set in `prefs.conf` in `recurring_schedule_time`, which defaults to 10 pm.

> Note: Once a card is entered for storage, the real account number can never be extracted again via the API. Only the card type and a truncated account number will be returned.

5.2.1 Add Token or Recurring Payment

Create a new token in the Card Storage subsystem, optionally also entering it into the Recurring Billing subsystem.

The various ways that card/account information can be sent to Monetra are the same as for a sale/authorization request and will not be reiterated below. For details on this, see Appendix A.1: Account Information.

If a Merchant User is part of a token group, then all tokens created will be added to that token group's pool, meaning any member of the token group can then use, edit, and delete the token. See Section 4.10.1: Manage Token Groups for more information on token groups.

Required and optional parameters for each token type:

	store	recurring	installment	nontrans
Account Data	Y	Y	Y	Y
amount		Y	Y	
frequency		Y	Y	
bdate		O	O	
edate		O		
installment_total			Y	
active	O	O	O	O

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringadd**
account data	Y		Multiple methods. See Appendix A.1: Account Information.
type	Y	A	Type of token. Possible values: store recurring installment nontrans
matching_token	O	A	Boolean flag indicating whether or not to search for tokens matching this account. If found, the token will be updated instead of a new token being added. Defaults to no.
amount	C	M	For recurring payments, amount to charge every payment. For installment payments, total amount to charge across all payments.
frequency	C	A	For recurring and installment tokens, frequency of billing. Possible values: daily - Every day weekly - Every week biweekly - Two times per month monthly - Monthly

Request Parameters			
Key	**Req**	**Spec**	**Description**
			bimonthly - Every other month quarterly - Four times per year semiannually - Two times per year annually - Every year
bdate	C	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	C	ANS	Ending date. See Appendix C.8: Date Formats.
installment_total	C	N, Max 2	For installment payments, the total number of payments to make
active	O	A	Boolean flag indicating whether or not to mark token as active. Defaults to yes.
descr	O	ANS, Max 2048	Free-form description
clientref	O	N, Max 19	Client reference number
customer_id	O	N, Max 19	Customer ID, as returned from the Customer Management subsystem

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
token	N, Max 19	Token ID

5.2.2 Edit Token or Recurring Payment

Edit an existing token's data in the Card Storage subsystem or a recurring payment in the Recurring Billing subsystem. The token must be in active status.

It is invalid to change the token type when editing an token. Instead, a new token should be created from the original, which is done by referencing the original token's number when creating the new one. The account data will be pulled from the original token and used for the new one. The same is true for the frequency and start date of recurring and installment tokens after the cycle has been started.

In addition to the parameters below, card/account information can be updated as per the parameters in Appendix A.1: Account Information.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringedit**
token	Y	N, Max 19	Token ID
account data	O		Multiple methods. See Appendix A.1: Account Information.
amount	O	M	For recurring payments, amount to charge every payment. For installment payments, total amount to charge across all payments.
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.
installment_total	O	N, Max 2	For installment payments, the total number of payments to make
active	O	A	Boolean flag indicating whether or not to mark token as active. Defaults to yes.
descr	O	ANS, Max 2048	Free-form description
clientref	O	N, Max 19	Client reference number
customer_id	O	N, Max 19	Customer ID, as returned from the Customer Management subsystem

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.2.3 Delete Token or Recurring Payment

Delete an existing token in the Card Storage subsystem and its associated data. If the token is set up for recurring payments, then those payments will stop.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringdel**
token	Y	N, Max 19	Token ID

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.2.4 List Tokens and Recurring Payments

Get a datablock of tokens for the Merchant User or token group, optionally limited by various parameters.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringlist**
account	O	N	Account number on card
token	O	N, Max 19	Token ID
active	O	A	Boolean flag indicating whether or not to reference active tokens. Defaults to yes.
expdate_end	O	N, 4 or 6	Last expiration date in the future to show, formatted as MMYY or YYYYMM
type	O	A	Type of token. Possible values: store recurring installment nontrans
clientref	O	N, Max 19	Client reference number
customer_id	O	N, Max 19	Unique ID assigned to customer
cardholdername	O	ANS	Name on card/check
cardtypes	O	AS	List of card types, separated by pipes (|). See Appendix C.1: Card Types.

Report Fields		
Key	**Spec**	**Description**
token	N, Max 19	Token ID
type	A	Type of token. Possible values: store

Report Fields		
Key	**Spec**	**Description**
		recurring installment nontrans
create_ts	NS	Timestamp of when token was created, YYYY-MM-DD HH:MM:SS +/-ZZZZ
update_ts	NS	Timestamp of when token was last updated, YYYY-MM-DD HH:MM:SS +/-ZZZZ
flags	AS	List of token flags, separated by pipes (|). Possible values: PROC_TOKEN - Processor token ACCT_BUSINESS - ACH business account ACCT_SAVINGS - ACH savings account
active	A	Boolean flag indicating whether or not the token is active, or done for completed installment tokens.
cardtype	A	Card type detected. See Appendix C.1: Card Types.
abaroute	N, 9	ABA routing number from check
account	N	Account number on card
expdate	N, 4	Expiration date printed on card, MMYY
cardholdername	ANS	Name on card/check
street	ANS	Billing street address for card
zip	AN	Zip code for AVS. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
descr	ANS, Max 2048	Free-form description
clientref	N, Max 19	Client reference number
customer_id	N, Max 19	Unique ID assigned to customer
customer_display_name	ANS	Display name of customer
amount	M	For recurring payments, amount to charge every payment. For installment payments, total amount to charge across all payments.
cof_transid	N	Transaction ID returned from the first COF transaction. Must be sent on all subsequent

Report Fields		
Key	**Spec**	**Description**
		COF transactions. Not needed when referencing a stored token.
cof_authamount	M	Authorized amount returned from the first COF transaction. Must be sent on all subsequent COF transactions. Not needed when referencing a stored token.
frequency	A	For recurring and installment tokens, frequency of billing. Possible values: daily - Every day weekly - Every week biweekly - Two times per month monthly - Monthly bimonthly - Every other month quarterly - Four times per year semiannually - Two times per year annually - Every year
bdate	ANS	For recurring and installment tokens, beginning date. See Appendix C.8: Date Formats.
edate	ANS	For installment tokens, ending date. See Appendix C.8: Date Formats.
installment_num	N, Max 2	For installment tokens, the current payment number in a series of installments
installment_total	N, Max 2	For installment tokens, the total number of payments to make
last_run_id	N	For recurring and installment tokens, database ID from recurring_hist table for last run
last_success_date	NS	For recurring and installment tokens, timestamp of last successful run, YYYY-MM-DD HH:MM:SS +/-ZZZZ
last_run_date	NS	For recurring and installment tokens, timestamp of last run, YYYY-MM-DD HH:MM:SS +/-ZZZZ
next_run_date	NS	For recurring and installment tokens, timestamp of next scheduled run, YYYY-MM-DD HH:MM:SS +/-ZZZZ
next_run_amount	M	For recurring and installment tokens, transaction amount for next scheduled run
billing_merchant	ANS	Username of merchant
desc_merch	ANS	Merchant name descriptor, meant to change how a merchant is displayed on a cardholder's

Report Fields		
Key	**Spec**	**Description**
		receipt. Formatting varies from processor to processor, and not all processors support this feature.
desc_loc	ANS	Merchant location descriptor. Formatting varies from processor to processor, and not all processors support this feature.
In addition to the parameters above, this request also returns any custom fields set for the merchant.		

5.2.5 Get Token Count

Get the number of tokens available to the Merchant User.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringlist**
recurringlist	Y		**count**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
count	N	Number of tokens available

5.2.6 List Recurring History

Get a datablock of transactions that were run from the Recurring Billing subsystem, optionally limited by a specific token, a specific transaction, or a date range. This report will only show transactions that were scheduled and used with a recurring or installment token.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringhist**
token	O	N, Max 19	Token ID
hist_id	O	N, Max 19	ID of history record
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.

Report Fields		
Key	**Spec**	**Description**
id	N, Max 19	ID of transaction for token
token	N, Max 19	Token ID
timestamp	NS	Timestamp for transaction, YYYY-MM-DD HH:MM:SS +/-ZZZZ
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.2.7 Clear Token History

Clear the token history of transactions that were run from the Recurring Billing subsystem. This affects only transactions that were scheduled and used with a recurring or installment token. To schedule this request on a periodic basis with Monetra's Cron subsystem, use cron_task=purgerecurhist.

This does not clear the transaction history; it clears the history of token usage, as viewed with the List Recurring History request. To clear the transaction history, see Section 5.7.1: Clear Transaction History.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringpurgehist**
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.2.8 Delete Expired Tokens

Remove all tokens from the Card Storage subsystem that have accounts with expired cards. This will only act on store and nontrans tokens. This can be automated on a periodic basis with Monetra's Cron subsystem using cron_task=purgeexpired. To instead update the account information for expired tokens, see Section 4.13: Account Updater.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringpurgeexpired**
keep_months	O	N	Number of months back to keep

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.3 Customer Management

Monetra includes a subsystem for managing customers. This allows an integrator to add customer and company details to the Monetra database--including multiple addresses per customer--and assign multiple DSS tokens to a single customer.

The customer database honors token groups for sharing of customers between accounts.

> Note: While there is a shared hierarchy of action names, the customer management subsystem is functionally distinct from the Card Storage and Recurring Billing subsystems and does not rely on tokens or recurring payments.

5.3.1 Add Customer

Add a new customer to the customer database for the merchant or token group.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`admin`**
`admin`	Y		**`recurringadd`**
`recurringadd`	Y		**`customer`**
`display_name`	Y	ANS, Max 128	Display name
`flags`	O	ANS	List of flags for customer, separated by pipes (|). Possible values: `taxexempt` - The customer is exempt from paying tax `email_receipt_recurring` - For recurring/installment transactions, email a copy of the receipt to the customer's email address
`name_company`	O	ANS, Max 128	Company name
`name_prefix`	O	ANS, Max 32	Name prefix (e.g. `Mr.`, `Mrs.`, `Ms.`, `Dr.`)
`name_first`	O	ANS, Max 32	First name

Request Parameters			
Key	**Req**	**Spec**	**Description**
name_middle	O	ANS, Max 32	Middle name
name_last	O	ANS, Max 32	Last name
name_suffix	O	ANS, Max 32	Name suffix (e.g. Jr., III)
phone_work	O	ANS, Max 32	Work phone number
phone_home	O	ANS, Max 32	Home phone number
phone_mobile	O	ANS, Max 32	Mobile phone number
phone_fax	O	ANS, Max 32	Fax number
email	O	ANS, Max 128	Email address
website	O	ANS, Max 128	Website URL
business_id	O	ANS, Max 32	Business ID (EIN/FEIN)
accounting_id	O	ANS, Max 128	ID used in external accounting system for customer
notes	O	ANS, Max 2048	Free-form field for general customer information
default_token	O	N, Max 19	Default token

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
id	N, Max 19	Unique ID assigned to customer

5.3.2 Edit Customer

Edit an existing customer.

After an address has been created, this action can be used to link that address to a customer.

Note: If the customer's token is part of a token group, then all fields specified in that token group's custom_customer_fields are valid parameters in addition to those listed below. If the token group does not have any custom fields, then all fields specified in the Merchant User's merch_customer_fields are valid.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringedit**
recurringedit	Y		**customer**
id	Y	N, Max 19	Unique ID assigned to customer
display_name	O	ANS, Max 128	Display name
name_company	O	ANS, Max 128	Company name

Request Parameters			
Key	**Req**	**Spec**	**Description**
name_prefix	O	ANS, Max 32	Name prefix (e.g. Mr., Mrs., Ms., Dr.)
name_first	O	ANS, Max 32	First name
name_middle	O	ANS, Max 32	Middle name
name_last	O	ANS, Max 32	Last name
name_suffix	O	ANS, Max 32	Name suffix (e.g. Jr., III)
phone_work	O	ANS, Max 32	Work phone number
phone_home	O	ANS, Max 32	Home phone number
phone_mobile	O	ANS, Max 32	Mobile phone number
phone_fax	O	ANS, Max 32	Fax number
email	O	ANS, Max 128	Email address
website	O	ANS, Max 128	Website URL
business_id	O	ANS, Max 32	Business ID (EIN/FEIN)
accounting_id	O	ANS, Max 128	ID used in external accounting system for customer

Request Parameters			
Key	**Req**	**Spec**	**Description**
notes	O	ANS, Max 2048	Free-form field for general customer information
default_billing_id	O	N, Max 19	ID of address to use as default billing address, as returned from Add Address
default_shipping_id	O	N, Max 19	ID of address to use as default shipping address, as returned from Add Address
default_token	O	N, Max 19	Default token

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.3.3 Delete Customer

Delete an existing customer.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringdel**
recurringdel	Y		**customer**
id	Y	N, Max 19	Unique ID assigned to customer

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.3.4 List Customers

Get a datablock of customers in the customer database and their associated information, optionally limited to a single provided customer.

The billing address and shipping address provided by the report are the customer's default addresses. They may not be present if no addresses were set as defaults.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringlist**
recurringlist	Y		**customer**
id	O	N, Max 19	Unique ID assigned to customer
name	O	ANS, Max 128	Display name
name_company	O	ANS, Max 128	Company name
phone	O	ANS, Max 32	One of the work, home, or mobile phone numbers
email	O	ANS, Max 128	Email address

Report Fields		
Key	**Spec**	**Description**
id	N, Max 19	Unique ID assigned to customer
ts_created	NS	Timestamp of when customer was created, YYYY-MM-DD HH:MM:SS +/-ZZZZ
ts_modified	NS	Timestamp of when customer was last updated, YYYY-MM-DD HH:MM:SS +/-ZZZZ
flags	ANS	List of flags for customer, separated by pipes (|). Possible values: taxexempt - The customer is exempt from paying tax email_receipt_recurring - For recurring/installment transactions, email a copy of the receipt to the customer's email address
display_name	ANS, Max 128	Display name
name_company	ANS, Max 128	Company name
name_prefix	ANS, Max 32	Name prefix (e.g. Mr., Mrs., Ms., Dr.)
name_first	ANS, Max 32	First name
name_middle	ANS, Max 32	Middle name
name_last	ANS, Max 32	Last name
name_suffix	ANS, Max 32	Name suffix (e.g. Jr., III)
phone_work	ANS, Max 32	Work phone number
phone_home	ANS, Max 32	Home phone number
phone_mobile	ANS, Max 32	Mobile phone number
phone_fax	ANS, Max 32	Fax number

Report Fields

Key	Spec	Description
email	ANS, Max 128	Email address
website	ANS, Max 128	Website URL
business_id	ANS, Max 32	Business ID (EIN/FEIN)
accounting_id	ANS, Max 128	ID used in external accounting system for customer
notes	ANS, Max 2048	Free-form field for general customer information
default_billing_id	N, Max 19	ID of address to use as default billing address, as returned from Add Address
default_shipping_id	N, Max 19	ID of address to use as default shipping address, as returned from Add Address
default_token	N, Max 19	Default token
billing_display_name	ANS, Max 128	Billing display name
billing_address1	ANS, Max 128	Billing address, line 1
billing_address2	ANS, Max 128	Billing address, line 2
billing_city	ANS, Max 128	Billing city or province
billing_state	ANS, Max 128	Billing state
billing_country	A, 3	3-letter billing country code (ISO 3166-1 alpha-3)
billing_postal_code	ANS, Max 32	Billing postal or zip code

Report Fields		
Key	**Spec**	**Description**
billing_delivery_notes	ANS, Max 2048	Delivery notes
shipping_display_name	ANS, Max 128	Shipping display name
shipping_address1	ANS, Max 128	Shipping address, line 1
shipping_address2	ANS, Max 128	Shipping address, line 2
shipping_city	ANS, Max 128	Shipping city or province
shipping_state	ANS, Max 128	Shipping state
shipping_country	A, 3	3-letter shipping country code (ISO 3166-1 alpha-3)
shipping_postal_code	ANS, Max 32	Shipping postal or zip code
shipping_delivery_notes	ANS, Max 2048	Delivery notes

5.3.5 Add Address

Add a new address to a customer.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringadd**
recurringadd	Y		**customeraddress**
customer_id	Y	N, Max 19	Unique ID assigned to customer

Request Parameters			
Key	**Req**	**Spec**	**Description**
display_name	O	ANS, Max 128	Display name
address1	Y	ANS, Max 128	Address, line 1
address2	O	ANS, Max 128	Address, line 2
city	O	ANS, Max 128	City or province
state	O	ANS, Max 128	State
country	O	A, 3	3-letter country code (ISO 3166-1 alpha-3)
postal_code	O	ANS, Max 32	Postal or zip code
delivery_notes	O	ANS, Max 2048	Delivery notes
default_billing	O	A	Boolean flag indicating whether or not this address will be the default billing address for the customer. Defaults to no.
default_shipping	O	A	Boolean flag indicating whether or not this address will be the default shipping address for the customer. Defaults to no.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
id	N, Max 19	Unique ID assigned to a customer's address

5.3.6 Edit Address

Edit an existing address.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringedit**
recurringedit	Y		**customeraddress**
customer_id	Y	N, Max 19	Unique ID assigned to customer
id	Y	N, Max 19	Unique ID assigned to a customer's address
display_name	O	ANS, Max 128	Display name
address1	O	ANS, Max 128	Address, line 1
address2	O	ANS, Max 128	Address, line 2

Request Parameters			
Key	**Req**	**Spec**	**Description**
city	O	ANS, Max 128	City or province
state	O	ANS, Max 128	State
country	O	A, 3	3-letter country code (ISO 3166-1 alpha-3)
postal_code	O	ANS, Max 32	Postal or zip code
delivery_notes	O	ANS, Max 2048	Delivery notes
default_billing	O	A	Boolean flag indicating whether or not this address will be the default billing address for the customer. Defaults to no.
default_shipping	O	A	Boolean flag indicating whether or not this address will be the default shipping address for the customer. Defaults to no.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.3.7 Delete Address

Delete an existing customer's address.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringdel**
recurringdel	Y		**customeraddress**
customer_id	Y	N, Max 19	Unique ID assigned to customer
id	Y	N, Max 19	Unique ID assigned to a customer's address

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.3.8 List Addresses

Get a datablock of addresses registered with the specified customer, optionally limited to a single provided address.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**recurringlist**
recurringlist	Y		**customeraddress**
customer_id	C	N, Max 19	Unique ID assigned to customer
id	C	N, Max 19	Unique ID assigned to a customer's address

Report Fields		
Key	**Spec**	**Description**
id	N, Max 19	Unique ID assigned to a customer's address
customer_id	N, Max 19	Unique ID assigned to customer
display_name	ANS, Max 128	Display name
address1	ANS, Max 128	Address, line 1
address2	ANS, Max 128	Address, line 2
city	ANS, Max 128	City or province
state	ANS, Max 128	State
country	A, 3	3-letter country code (ISO 3166-1 alpha-3)
postal_code	ANS, Max 32	Postal or zip code
delivery_notes	ANS, Max 2048	Delivery notes

5.4 Transaction Management

5.4.1 Edit Transaction Details

Edit the details of a transaction, as stored in Monetra.

This can edit many fields, such as those that affect interchange (e.g. tax, rate, bdate/edate), or user-definable data (e.g. comments, clerkid, stationid). This action will only alter transactions that are currently unsettled. Additionally, if a processor is host-based (retains control of the transaction on their end), then only uncaptured transactions are subject to edit.

Note: Generally, an **adjust** is the appropriate action for modifying a transaction.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**fieldedit**
ttid	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
In addition to the fields above, any parameter passed in the original transaction may also be sent.			

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.4.2 Force Void

Delete a previously-authorized transaction.

In the vast majority of cases, a **`reversal`** is the appropriate action, because that will remove the transaction from the batch *and* release the hold on funds.

For most processors, this is not a real-time action; the transaction is removed from the batch and therefore never sent for settlement, but the void is not typically sent online. This means that any hold on funds will *not* be removed. The customer will not be charged the amount, but the authorization will stay on the account until the issuer removes it, which is usually around a week.

The main use for a **`void`** is to ensure that the transaction will not settle if a **`reversal`** was issued but failed.

`Access level:`	Section 2.4.2: Merchant User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`admin`**
`admin`	Y		**`forcevoid`**
`ttid`	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.5 Batch Management

The actions outlined below provide a means to manage specific, administrative-level features of the batch management subsystem.

5.5.1 Set Batch Number

Set batch numbering to the given value.

This will have two effects on the batch number: it will change the open batch's number to the provided value, and it will set the batch counter to that same value. When the batch is then closed/settled, the batch number will be incremented for the new batch. For example, an open batch is numbered "10". When that batch is settled, the next batch will be numbered "11". If instead batch "10" has its number set to "50" using this action, then the next batch will be numbered "51". To only change a batch's number without affecting the numbering of any other batch, see Section 5.5.2: Renumber Single Batch.

This action is relevant when batch numbers conflict, such as can happen when a merchant is attempting to fix a settlement issue with a processor.

Some notes to keep in mind when modifying a batch number:

Processors keep batch numbers for 7 days to check for duplicates. Batch numbers within that window should not be reused.

It is probably wise to not reuse batch numbers within a 90-day window, so as to stay within the card brand's chargeback window to avoid an actual chargeback.

Monetra will retain data on a settled batch until that batch number is used again, at which point it will be overwritten.

`Access level:`	Section 2.4.2: Merchant User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`admin`**
`admin`	Y		**`setbatchnum`**
`batch`	Y	N, Max 10	Batch number
`sub`	C	N, Max 10	Subaccount to use (for split routes), or `0` to use the default route

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.5.2 Renumber Single Batch

Change an individual batch's number.

This will not affect any other batch number or the batch counter. For example, an open batch is numbered `10`. When that batch is settled, the next batch will be numbered `11`. If batch `10` has its number set to `50` using this action, then the next batch will still be numbered `11`. To also set the batch counter so that the next batch number will be incremented from the new batch number provided (to `50` in this example), see Section 5.5.1: Set Batch Number.

This action is relevant when batch numbers conflict, such as can happen when a merchant is attempting to fix a settlement issue with a processor.

Some notes to keep in mind when modifying a batch number:

Processors keep batch numbers for 7 days to check for duplicates. Batch numbers within that window should not be reused.

It is probably wise to not reuse batch numbers within a 90-day window, so as to stay within the card brand's chargeback window to avoid an actual chargeback.

Monetra will retain data on a settled batch until that batch number is used again, at which point it will be overwritten.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**renumberbatch**
batch	Y	N, Max 10	Batch number
newbatch	Y	N	New batch number. Maximum value depends on processor.
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.5.3 Close Batch

Close the currently open batch *without* sending it to the processor.

This will force Monetra to start adding transactions to a new batch.

The closed batch will remain in an unsettled state and will not be sent to the processor. To settle a batch, which will send it to the processor to request funds, see Section 6.5.1: Settle Batch. Typically, this is handled on a daily basis using the Cron subsystem.

Only one batch per settlement subaccount can be open at a time. This request enables a Merchant User to close a batch so as to prevent additional transactions from being added to it while preventing the batch from being sent online to the processor.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**closebatch**
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.5.4 Force-Settle Batch

Mark a batch as settled within Monetra *without* sending it online to the processor or requesting funding.

This should only be used if the processor has accepted the batch, but Monetra did not receive the response.

Note: This request should be used with extreme caution. If the processor has not received the batch, then force-settling it will prevent funding of those transactions in the batch.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**forcesettle**
batch	Y	N, Max 10	Batch number
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route
inquiry	O	A	Boolean flag indicating whether or not to do a dry run. Defaults to no.
comments	O	ANS	Free-form comment field
clerkid	O	ANS	Clerk ID

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.5.5 Unsettle Batch

Mark a previously-settled batch as unsettled within Monetra.

This will allow the batch to be submitted for funding again. A batch should only be unsettled when the processor has confirmed that they did not receive the batch but it is showing as settled in Monetra.

Note: This action should only be taken with confirmation from the processor that this should happen. Unsettling and then settling again can result in double charging customers.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`admin`**
`admin`	Y		**`unsettlebatch`**
`batch`	Y	N, Max 10	Batch number
`sub`	C	N, Max 10	Subaccount to use (for split routes), or `0` to use the default route

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.6 Reports

This section provides information on the various reports that can be generated from Monetra.

5.6.1 Get Raw Data

Get a datablock of the raw data as returned from the processing institution for all unsettled transactions, optionally limited by the request parameters below.

Note: This reporting feature is licensed separately. To take advantage of these features, you might be required to upgrade.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**interchange**
ttid	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
bdate	C	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	C	ANS	Ending date. See Appendix C.8: Date Formats.
cardtypes	O	AS	List of card types, separated by pipes (|). See Appendix C.1: Card Types.
batch	O	N, Max 10	Batch number

Report Fields		
Key	**Spec**	**Description**
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
last4	N, Fixed 4	Last 4 digits of the account number
account	N	Account number on card
abaroute	N, 9	ABA routing number from check
amount	M	Amount of transaction
reqamount	M	Requested amount

Report Fields

Key	Spec	Description
orig_authamount	M	authamount from the original transaction
authamount	M	Amount actually authorized. Only returned if less than requested amount
cashbackamount	M	Amount of total given to customer as cash. Required if customer is receiving cash.
cardtype	A	Card type detected. See Appendix C.1: Card Types.
entrymode	A, Fixed 1	Monetra-defined value indicating how the transaction data was entered. See Appendix B.4: Alphabetical Listing.
divisionnum	ANS	Description of division within company, usually for ACH transactions
descmerch	ANS	Merchant name descriptor, meant to change how a merchant is displayed on a cardholder's receipt. Formatting varies from processor to processor, and not all processors support this feature.
descloc	ANS	Merchant location descriptor. Formatting varies from processor to processor, and not all processors support this feature.
custref	ANS, Max 128	Customer Reference Number
batch	N, Max 10	Batch number
expdate	N, 4	Expiration date printed on card, MMYY
cardholdername	ANS	Name on card/check
item	N	Sequence number
installment_num	N, Max 2	For installment payments, the current payment number in a series of installments
installment_total	N, Max 2	For installment payments, the total number of payments to make
ordernum	ANS	Order number
proc	ANS	Name of processor used
tax	M	Amount of total that is tax applied to the order, or nt if the transaction is non-taxable/ tax-exempt
tip	M	Amount of total that went towards the tip, as sent in the examount field

Report Fields		
Key	**Spec**	**Description**
type	A	Transaction Type, as outlined in Chapter 6: *Merchant Subuser Actions*
tranflags	ANS	Monetra-defined transaction flags. See Appendix B.5.18: Transaction Flags.
ts	N, Max 19	For Debit/EBT, Authorization timestamp, in seconds since the Unix epoch
zip	AN	Zip code for AVS. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
actcode	ANS	Action Code. See Appendix B.5.1: Action Code.
checknum	N	Sequential number on check
accounttype	AS	Type of bank account. See Appendix A.1.6: Bank Account (ACH).
apcode	AN	Approval Code. See Appendix B.5.2: Approval Code.
authsource	AN, Fixed 1	Authorization Source Code. See Appendix B.5.3: Authorization Source Code.
avsresp	AN, Fixed 1	Raw code from processor indicating AVS result. See Appendix B.5.4: AVS Response Code.
cardlevel	AN, Max 2	Card level result. See Appendix B.3: Card Level Result Codes.
cavvresp	AN, Fixed 1	Raw code from processor indicating CAVV result. See Appendix B.5.6: CAVV Result Code.
commind	AN, Fixed 1	Commercial Card Response Indicator. See Appendix B.5.7: Commercial Card Response Indicator.
cvresp	A	Raw code from processor indicating CV result. See Appendix B.5.8: Cardholder Verification Response Code.
netident	AN, Fixed 1	Network Identification Code. See Appendix B.5.10: Network Identification Code.
posdata	ANS	POS Data. See Appendix B.5.11: POS Data Code.
proccode	AN, Fixed 6	Processing Code. See Appendix B.5.12: Processing Code.

Report Fields		
Key	**Spec**	**Description**
raci	AN, Fixed 1	Return Authorization Characteristics Indicator. See Appendix B.5.13: Returned Authorization Characteristics Indicator.
rrefnum	AN, Max 12	Retrieval Reference Number. See Appendix B.5.14: Retrieval Reference Number.
settledate	N, Fixed 4	Settlement Date as MMDD, as returned by the processor. Debit/EBT cards only.
stan	AN	System Trace Audit Number. See Appendix B.5.16: System Trace Audit Number.
trandate	N, Fixed 6	Processor-set transaction date, MMDDYY
transid	ANS, Fixed 15	Transaction ID. See Appendix B.5.17: Transaction ID.
trantime	N, Fixed 6	Processor-set transaction time, HHMMSS
valcode	AN, Fixed 1	Validation Code. See Appendix B.5.19: Validation Code.
discountamount	M	Amount of discount applied to the order as a whole. If Level III line items are part of the order, this includes all discounts for each line item. The sum of all line item discounts must be less than or equal to this value.
freightamount	M	Amount of freight/shipping on the order. For Level III, if shipping was charged on the order, this field must be sent. If the charge was greater than 0, then shipzip must also be sent. Freight is never part of the individual line items.
dutyamount	M	Total amount of duty (fee associated with import of goods) on order
shipzip	AN	Zip code where product is being shipped. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
shipcountry	AN	Country where product is being shipped, formatted as ISO 3166-1 numeric or alpha-3
l3num	N	Number of Level III line items
bdate	ANS	Beginning date. See Appendix C.8: Date Formats.

Report Fields		
Key	**Spec**	**Description**
edate	ANS	Ending date. See Appendix C.8: Date Formats.
roomnum	ANS	For Lodging, room number
excharges	AS	For Lodging, additional charges not included in the room rate. See Appendix C.7: Extra Charge Codes.
rate	M	For Lodging, room rate per night
sqi	AN	Spend Qualified Indicator (SQI) or Transaction Integrity Class (TIC). See Appendix B.5.15: Spend Qualified Indicator.
cavv	ANS, Max 40	3-D Secure data. Base64-encoded CAVV/AAV response data from the VISA/MC authentication servers. If the card issuer does not use this system, the card issuer does but the cardholder does not, or the system is currently unavailable, this is nonparticipant. Additionally, if the ECI is known (typically as part of the mobile in-app payload data), it is prefixed to the CAVV data and pipe (|) separated.
icc	X	TLV data, as returned from an EMV device for a chip insert or tap
laneid	N, Max 8	Unique lane/register identifier
surchargeamount	M	Amount of total being received as a surcharge
freighttaxamount	M	VAT/Tax amount for freight
freighttaxrate	M	VAT/Tax tax for freight
nationaltax	M	For European markets, national sales tax
merchvatnum	AN	Merchant's VAT number
custvatnum	ANS	Customer VAT registration number
othertax	M	For European markets, other tax
vatinvoicenum	ANS	For European markets, VAT Invoice Number
summarycommoditycode	ANS	For European markets, used to categorize purchases for VAT reporting
healthamount	M	Health amount
rxamount	M	For healthcare, amount of total that is for prescriptions
clinicamount	M	For healthcare, amount of total that is for clinic-related services

Report Fields		
Key	**Spec**	**Description**
visionamount	M	For healthcare, amount of total that is for vision-related services
dentalamount	M	For healthcare, amount of total that is for dental-related services
transitamount	M	Amount of total that is for transportation
3ds_txnid	ANS	EMV 3-D Secure 2.0 Transaction ID generated by the Directory Server
wallet_identifier	ANS, Max 3	For Mastercard, Wallet Identifier
cof_authamount	M	Authorized amount returned from the first COF transaction. Must be sent on all subsequent COF transactions. Not needed when referencing a stored token.
cof_transid	N	Transaction ID returned from the first COF transaction. Must be sent on all subsequent COF transactions. Not needed when referencing a stored token.
esi	ANS, Fixed 3	Ecommerce Security Indicator, for Mastercard 3-D Secure transactions
token_requestor	ANS, Max 11	For Mastercard, Token Requestor
token_assurance_level	AN, Max 2	For Mastercard, Token Assurance Level
In addition to the fields above, this report also returns any custom fields set for the merchant. See Section 5.1.8: Get Merchant Information.		

5.6.2 Get Unsettled Transactions

Get a datablock of all unsettled transactions for the current Merchant User, matching on provided parameters. This is useful for monitoring which transactions have yet to be settled. This report is very flexible and can be modified with a large number of parameters, as detailed below.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**gut**

Request Parameters			
Key	**Req**	**Spec**	**Description**
`user`	O	ANS, Max 256	Merchant User that ran the transaction
`capture`	O	A	Capture status. If not sent, both captured and uncaptured transactions will be shown. Possible values: `yes` - Show only captured transactions `no` - Show only uncaptured transactions
`ttid`	O	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
`bdate`	O	ANS	Beginning date. See Appendix C.8: Date Formats.
`edate`	O	ANS	Ending date. See Appendix C.8: Date Formats.
`last_modified_bdate`	O	N	Used to filter report by date range. This is the beginning date.
`last_modified_edate`	O	N	Used to filter report by date range. This is the ending date.
`type`	O	AS	List of transaction types to match, separated by pipes (`|`). Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
`cardtypes`	O	AS	List of card types, separated by pipes (`|`). See Appendix C.1: Card Types.
`sub`	C	N, Max 10	Subaccount to use (for split routes), or `0` to use the default route
`batch`	O	N, Max 10	Batch number
`pclevel`	O	N	Card level. Possible values: `0` - Non-commercial card `1` - Commercial card `2` - Purchase card
`acct`	O	N, Min 4	Account number. Note: It is valid to pass in only the last four digits of the card.
`clerkid`	O	ANS	Clerk ID

Request Parameters			
Key	**Req**	**Spec**	**Description**
stationid	O	N, Max 10	Station ID
comments	O	ANS	Free-form comment field
cardholdername	O	ANS	Name on card/check
amount	O	M	Amount of transaction

Report Fields		
Key	**Spec**	**Description**
user	ANS, Max 256	Merchant User that ran the transaction
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
type	A	Transaction type. Possible transaction types are any listed in Chapter 6: *Merchant Subuser Actions*.
proc	ANS	Name of processor used
entrymode	A, Fixed 1	Monetra-defined value indicating how the transaction data was entered. See Appendix B.4: Alphabetical Listing.
txnstatus	A	List of transaction status flags, separated by pipes (|). Possible values: DECLINED - Transaction was declined UNCAPTURED - Transaction is not captured (i.e. not added to a batch for later settlement) CAPTURED - Transaction was captured (i.e. added to the batch for later settlement) COMPLETE - Transaction is complete (i.e. settled or voided) VOIDED - Transaction was reversed/voided ONLINE - Transaction went out to the processing institution

Report Fields		
Key	**Spec**	**Description**
		SENSITIVEDATA - Transaction still contains sensitive data on file
tranflags	ANS	Monetra-defined transaction flags. See Appendix B.5.18: Transaction Flags.
capture	A	Boolean flag indicating whether or not the transaction was captured (i.e. added to the batch for later settlement)
card	A	Card type detected. See Appendix C.1: Card Types.
pclevel	N	Card level. Possible values: 0 - Non-commercial card 1 - Commercial card 2 - Purchase card
cardlevel	AN, Max 2	Card level result. See Appendix B.3: Card Level Result Codes.
abaroute	N, 9	ABA routing number from check
account	AN	Masked account number
expdate	N, 4	Expiration date printed on card, MMYY
checknum	N	Sequential number on check
timestamp	NS	Timestamp for transaction, YYYY-MM-DD HH:MM:SS +/-ZZZZ
last_modified_ts	NS	Timestamp of last transaction modification, YYYY-MM-DD HH:MM:SS +/-ZZZZ
accounttype	AS	Type of bank account. See Appendix A.1.6: Bank Account (ACH).
amount	M	Amount of transaction
reqamount	M	Requested amount
orig_authamount	M	authamount from the original transaction
authamount	M	Amount actually authorized
examount	M	Extra amount. Typically used for Retail and Restaurant tipping.
tax	M	Amount of total that is tax applied to the order, or nt if the transaction is non-taxable/ tax-exempt
cashback	M	Amount of total given to customer as cash. Required if customer is receiving cash.
authnum	AN	Approval Code. See Appendix B.5.2: Approval Code.

Report Fields

Key	Spec	Description
stan	AN	System Trace Audit Number. See Appendix B.5.16: System Trace Audit Number.
batch	N, Max 10	Batch number
item	N	Sequence number
cardholdername	ANS	Name on card/check
avs	A	Address verification result. See Appendix B.2.1: AVS Result Codes.
cv	A	Cardholder verification result. See Appendix B.2.2: CV Result Codes.
clerkid	ANS	Clerk ID
stationid	N, Max 10	Station ID
comments	ANS	Free-form comment field
divisionnum	ANS	Description of division within company, usually for ACH transactions
promoid	N	Promotion ID. Primarily for BillMeLater
descmerch	ANS	Merchant name descriptor, meant to change how a merchant is displayed on a cardholder's receipt. Formatting varies from processor to processor, and not all processors support this feature.
descloc	ANS	Merchant location descriptor. Formatting varies from processor to processor, and not all processors support this feature.
ptrannum	N, Max 19	Similar to ordernum, but numeric for efficient indexing in the database. It is recommended to use ordernum instead.
ordernum	ANS	Order number
custref	ANS, Max 128	Customer Reference Number
discountamount	M	Amount of discount applied to the order as a whole. If Level III line items are part of the order, this includes all discounts for each line item. The sum of all line item discounts must be less than or equal to this value.
freightamount	M	Amount of freight/shipping on the order. For Level III, if shipping was charged on the

Report Fields		
Key	**Spec**	**Description**
		order, this field must be sent. If the charge was greater than 0, then shipzip must also be sent. Freight is never part of the individual line items.
dutyamount	M	Total amount of duty (fee associated with import of goods) on order
shipzip	AN	Zip code where product is being shipped. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
shipcountry	AN	Country where product is being shipped, formatted as ISO 3166-1 numeric or alpha-3
l3num	N	Number of Level III line items
bdate	ANS	For Lodging, beginning date of stay. See Appendix C.8: Date Formats.
edate	ANS	For Lodging, ending date of stay. See Appendix C.8: Date Formats.
roomnum	ANS	For Lodging, room number
excharges	AS	For Lodging, additional charges not included in the room rate. See Appendix C.7: Extra Charge Codes.
rate	M	For Lodging, room rate per night
raw_code	AN	If raw codes are enabled, code from processor indicating result of transaction. Each processor has different specifications for this value.
raw_avs	AN, Fixed 1	If raw codes are enabled, code from processor indicating AVS result. See Appendix B.5.4: AVS Response Code.
raw_cv	A, Fixed 1	If raw codes are enabled, code from processor indicating CV result. See Appendix B.5.8: Cardholder Verification Response Code.
raw_cavv	A, Fixed 1	If raw codes are enabled, code from processor indicating CAVV result. See Appendix B.5.6: CAVV Result Code.
raw_cardlevel	ANS, Fixed 2	If raw codes are enabled, code from processor indicating card level according to Visa 62.23 standard. See Appendix B.3: Card Level Result Codes.
customer_id	N, Max 19	Unique ID assigned to customer

Report Fields		
Key	**Spec**	**Description**
token	N, Max 19	Token ID
order_id	N, Max 19	Order ID associated with order from product/ order system
surchargeamount	M	Amount of total being received as a surcharge
In addition to the fields above, this report also returns any custom fields set for the merchant. See Section 5.1.8: Get Merchant Information.		

5.6.3 Get Settled Transactions

Get a datablock of settled transactions for the current Merchant User, matching on provided parameters. If no dates are specified and neither ttid nor batch are sent, then the report will only return data from the last 30 days.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**gl**
user	O	ANS, Max 256	Merchant User that ran the transaction
showvoids	O	A	Void status. Possible values: yes - Include voids (default) no - Don't include voids only - Show only voids
reversible	O	A	Reversible status. If not sent, both types will be shown. Possible values: yes - Show only transactions that can be reversed or unsettled no - Show only transactions that cannot be reversed or unsettled both - Show both types
ttid	O	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.

Request Parameters			
Key	**Req**	**Spec**	**Description**
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.
last_modified_bdate	O	N	Used to filter report by date range. This is the beginning date.
last_modified_edate	O	N	Used to filter report by date range. This is the ending date.
type	O	AS	List of transaction types to match, separated by pipes (|). Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
cardtypes	O	AS	List of card types, separated by pipes (|). See Appendix C.1: Card Types.
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route
batch	O	N, Max 10	Batch number
pclevel	O	N	Card level. Possible values: 0 - Non-commercial card 1 - Commercial card 2 - Purchase card
acct	O	N, Min 4	Account number. Note: It is valid to pass in only the last four digits of the card.
clerkid	O	ANS	Clerk ID
stationid	O	N, Max 10	Station ID
comments	O	ANS	Free-form comment field
cardholdername	O	ANS	Name on card/check
amount	O	M	Amount of transaction

Report Fields		
Key	**Spec**	**Description**
user	ANS, Max 256	Merchant User that ran the transaction

Report Fields		
Key	**Spec**	**Description**
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
type	A	Transaction type. Possible transaction types are any listed in Chapter 6: *Merchant Subuser Actions*.
proc	ANS	Name of processor used
entrymode	A, Fixed 1	Monetra-defined value indicating how the transaction data was entered. See Appendix B.4: Alphabetical Listing.
txnstatus	A	List of transaction status flags, separated by pipes (|). Possible values: DECLINED - Transaction was declined UNCAPTURED - Transaction is not captured (i.e. not added to a batch for later settlement) CAPTURED - Transaction was captured (i.e. added to the batch for later settlement) COMPLETE - Transaction is complete (i.e. settled or voided) VOIDED - Transaction was reversed/voided ONLINE - Transaction went out to the processing institution SENSITIVEDATA - Transaction still contains sensitive data on file
tranflags	ANS	Monetra-defined transaction flags. See Appendix B.5.18: Transaction Flags.
reversible	A	Boolean flag indicating whether or not the transaction can be reversed or unsettled
card	A	Card type detected. See Appendix C.1: Card Types.
pclevel	N	Card level. Possible values: 0 - Non-commercial card 1 - Commercial card

Report Fields		
Key	**Spec**	**Description**
		2 - Purchase card
cardlevel	AN, Max 2	Card level result. See Appendix B.3: Card Level Result Codes.
abaroute	N, 9	ABA routing number from check
account	AN	Masked account number
expdate	N, 4	Expiration date printed on card, MMYY
checknum	N	Sequential number on check
timestamp	NS	Timestamp for transaction, YYYY-MM-DD HH:MM:SS +/-ZZZZ
last_modified_ts	NS	Timestamp of last transaction modification, YYYY-MM-DD HH:MM:SS +/-ZZZZ
accounttype	AS	Type of bank account. See Appendix A.1.6: Bank Account (ACH).
reasoncode	A	Reason for reversal/void. Possible values: REVERSAL_CLERKVOID - Clerk requested to void the transaction (default) REVERSAL_CUSTOMERCANCEL - Customer canceled transaction after approval (only if customer hits cancel) REVERSAL_DELIVERYFAILURE - Response could not be delivered to the POS REVERSAL_DEVICEFAILURE - Could not communicate with EMV terminal REVERSAL_CARDREMOVED - Card was removed prior to processor response REVERSAL_CARDDECLINE - Chip card declined the transaction after it was approved online REVERSAL_MACFAILURE - Interac MAC verification failure REVERSAL_FRAUD - Fraud suspected (default for fraudautodeny transactions) DECLINE_CHIPMALFUNCTION - EMV device was not able to read chip DECLINE_BADPIN - Incorrect PIN DECLINE_CARDDECLINE - Chip card declined the transaction before going online to the processor
amount	M	Amount of transaction
reqamount	M	Requested amount
orig_authamount	M	authamount from the original transaction

Report Fields		
Key	**Spec**	**Description**
authamount	M	Amount actually authorized
examount	M	Extra amount. Typically used for Retail and Restaurant tipping.
tax	M	Amount of total that is tax applied to the order, or nt if the transaction is non-taxable/ tax-exempt
cashback	M	Amount of total given to customer as cash. Required if customer is receiving cash.
authnum	AN	Approval Code. See Appendix B.5.2: Approval Code.
stan	AN	System Trace Audit Number. See Appendix B.5.16: System Trace Audit Number.
batnum	N, Max 10	Batch number
item	N	Sequence number
cardholdername	ANS	Name on card/check
avs	A	Address verification result. See Appendix B.2.1: AVS Result Codes.
cv	A	Cardholder verification result. See Appendix B.2.2: CV Result Codes.
clerkid	ANS	Clerk ID
stationid	N, Max 10	Station ID
comments	ANS	Free-form comment field
divisionnum	ANS	Description of division within company, usually for ACH transactions
promoid	N	Promotion ID. Primarily for BillMeLater
descmerch	ANS	Merchant name descriptor, meant to change how a merchant is displayed on a cardholder's receipt. Formatting varies from processor to processor, and not all processors support this feature.
descloc	ANS	Merchant location descriptor. Formatting varies from processor to processor, and not all processors support this feature.
ptrannum	N, Max 19	Similar to ordernum, but numeric for efficient indexing in the database. It is recommended to use ordernum instead.

Report Fields

Key	Spec	Description
ordernum	ANS	Order number
custref	ANS, Max 128	Customer Reference Number
discountamount	M	Amount of discount applied to the order as a whole. If Level III line items are part of the order, this includes all discounts for each line item. The sum of all line item discounts must be less than or equal to this value.
freightamount	M	Amount of freight/shipping on the order. For Level III, if shipping was charged on the order, this field must be sent. If the charge was greater than 0, then shipzip must also be sent. Freight is never part of the individual line items.
dutyamount	M	Total amount of duty (fee associated with import of goods) on order
shipzip	AN	Zip code where product is being shipped. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
shipcountry	AN	Country where product is being shipped, formatted as ISO 3166-1 numeric or alpha-3
l3num	N	Number of Level III line items
bdate	ANS	For Lodging, beginning date of stay. See Appendix C.8: Date Formats.
edate	ANS	For Lodging, ending date of stay. See Appendix C.8: Date Formats.
roomnum	ANS	For Lodging, room number
excharges	AS	For Lodging, additional charges not included in the room rate. See Appendix C.7: Extra Charge Codes.
rate	M	For Lodging, room rate per night
raw_code	AN	If raw codes are enabled, code from processor indicating result of transaction. Each processor has different specifications for this value.
raw_avs	AN, Fixed 1	If raw codes are enabled, code from processor indicating AVS result. See Appendix B.5.4: AVS Response Code.

Report Fields

Key	Spec	Description
raw_cv	A, Fixed 1	If raw codes are enabled, code from processor indicating CV result. See Appendix B.5.8: Cardholder Verification Response Code.
raw_cavv	A, Fixed 1	If raw codes are enabled, code from processor indicating CAVV result. See Appendix B.5.6: CAVV Result Code.
raw_cardlevel	ANS, Fixed 2	If raw codes are enabled, code from processor indicating card level according to Visa 62.23 standard. See Appendix B.3: Card Level Result Codes.
customer_id	N, Max 19	Unique ID assigned to customer
token	N, Max 19	Token ID
order_id	N, Max 19	Order ID associated with order from product/ order system
surchargeamount	M	Amount of total being received as a surcharge

In addition to the fields above, this report also returns any custom fields set for the merchant. See Section 5.1.8: Get Merchant Information.

5.6.4 Get Failed Transactions

Get a datablock of failed transactions for the current Merchant User, matching on provided parameters. If no dates are specified and neither ttid nor batch are sent, then the report will only return data from the last 30 days.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters

Key	Req	Spec	Description
action	Y		**admin**
admin	Y		**gft**
user	O	ANS, Max 256	Merchant User that ran the transaction
ttid	O	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.

Request Parameters			
Key	**Req**	**Spec**	**Description**
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.
last_modified_bdate	O	N	Used to filter report by date range. This is the beginning date.
last_modified_edate	O	N	Used to filter report by date range. This is the ending date.
type	O	AS	List of transaction types to match, separated by pipes (|). Available transactions are any listed in Chapter 6: *Merchant Subuser Actions*.
cardtypes	O	AS	List of card types, separated by pipes (|). See Appendix C.1: Card Types.
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route
batch	O	N, Max 10	Batch number
pclevel	O	N	Card level. Possible values: 0 - Non-commercial card 1 - Commercial card 2 - Purchase card
acct	O	N, Min 4	Account number. Note: It is valid to pass in only the last four digits of the card.
clerkid	O	ANS	Clerk ID
stationid	O	N, Max 10	Station ID
comments	O	ANS	Free-form comment field
cardholdername	O	ANS	Name on card/check
amount	O	M	Amount of transaction

Report Fields		
Key	**Spec**	**Description**
user	ANS, Max 256	Merchant User that ran the transaction

Report Fields		
Key	**Spec**	**Description**
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
type	A	Transaction type. Possible transaction types are any listed in Chapter 6: *Merchant Subuser Actions*.
proc	ANS	Name of processor used
entrymode	A, Fixed 1	Monetra-defined value indicating how the transaction data was entered. See Appendix B.4: Alphabetical Listing.
txnstatus	A	List of transaction status flags, separated by pipes (|). Possible values: DECLINED - Transaction was declined UNCAPTURED - Transaction is not captured (i.e. not added to a batch for later settlement) CAPTURED - Transaction was captured (i.e. added to the batch for later settlement) COMPLETE - Transaction is complete (i.e. settled or voided) VOIDED - Transaction was reversed/voided ONLINE - Transaction went out to the processing institution SENSITIVEDATA - Transaction still contains sensitive data on file
tranflags	ANS	Monetra-defined transaction flags. See Appendix B.5.18: Transaction Flags.
card	A	Card type detected. See Appendix C.1: Card Types.
abaroute	N, 9	ABA routing number from check
account	AN	Masked account number
expdate	N, 4	Expiration date printed on card, MMYY
checknum	N	Sequential number on check

Report Fields		
Key	**Spec**	**Description**
timestamp	NS	Timestamp for transaction, YYYY-MM-DD HH:MM:SS +/-ZZZZ
accounttype	AS	Type of bank account. See Appendix A.1.6: Bank Account (ACH).
reasoncode	A	Reason for reversal/void. Possible values: REVERSAL_CLERKVOID - Clerk requested to void the transaction (default) REVERSAL_CUSTOMERCANCEL - Customer canceled transaction after approval (only if customer hits cancel) REVERSAL_DELIVERYFAILURE - Response could not be delivered to the POS REVERSAL_DEVICEFAILURE - Could not communicate with EMV terminal REVERSAL_CARDREMOVED - Card was removed prior to processor response REVERSAL_CARDDECLINE - Chip card declined the transaction after it was approved online REVERSAL_MACFAILURE - Interac MAC verification failure REVERSAL_FRAUD - Fraud suspected (default for fraudautodeny transactions) DECLINE_CHIPMALFUNCTION - EMV device was not able to read chip DECLINE_BADPIN - Incorrect PIN DECLINE_CARDDECLINE - Chip card declined the transaction before going online to the processor
amount	M	Amount of transaction
batch	N, Max 10	Batch number
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
cardholdername	ANS	Name on card/check
avs	A	Address verification result. See Appendix B.2.1: AVS Result Codes.
cv	A	Cardholder verification result. See Appendix B.2.2: CV Result Codes.

Report Fields		
Key	**Spec**	**Description**
clerkid	ANS	Clerk ID
stationid	N, Max 10	Station ID
comments	ANS	Free-form comment field
divisionnum	ANS	Description of division within company, usually for ACH transactions
promoid	N	Promotion ID. Primarily for BillMeLater
descmerch	ANS	Merchant name descriptor, meant to change how a merchant is displayed on a cardholder's receipt. Formatting varies from processor to processor, and not all processors support this feature.
descloc	ANS	Merchant location descriptor. Formatting varies from processor to processor, and not all processors support this feature.
ptrannum	N, Max 19	Similar to ordernum, but numeric for efficient indexing in the database. It is recommended to use ordernum instead.
ordernum	ANS	Order number
custref	ANS, Max 128	Customer Reference Number
bdate	ANS	For Lodging, beginning date of stay. See Appendix C.8: Date Formats.
edate	ANS	For Lodging, ending date of stay. See Appendix C.8: Date Formats.
roomnum	ANS	For Lodging, room number
excharges	AS	For Lodging, additional charges not included in the room rate. See Appendix C.7: Extra Charge Codes.
rate	M	For Lodging, room rate per night
raw_code	AN	If raw codes are enabled, code from processor indicating result of transaction. Each processor has different specifications for this value.
raw_avs	AN, Fixed 1	If raw codes are enabled, code from processor indicating AVS result. See Appendix B.5.4: AVS Response Code.

Report Fields		
Key	**Spec**	**Description**
raw_cv	A, Fixed 1	If raw codes are enabled, code from processor indicating CV result. See Appendix B.5.8: Cardholder Verification Response Code.
raw_cavv	A, Fixed 1	If raw codes are enabled, code from processor indicating CAVV result. See Appendix B.5.6: CAVV Result Code.
raw_cardlevel	ANS, Fixed 2	If raw codes are enabled, code from processor indicating card level according to Visa 62.23 standard. See Appendix B.3: Card Level Result Codes.
customer_id	N, Max 19	Unique ID assigned to customer
token	N, Max 19	Token ID
order_id	N, Max 19	Order ID associated with order from product/ order system
In addition to the fields above, this report also returns any custom fields set for the merchant. See Section 5.1.8: Get Merchant Information.		

5.6.5 Get Transaction Details

Get the original transaction response, including the relevant data necessary for receipt printing.

Some of the response parameters for this action are unique, but most are common to all Merchant Subuser actions, like **sale**, **reversal**, etc. The parameters unique to **trandetail** are shown below. For details on other parameters, see the listings in Appendix A: *Request Parameters* and Appendix B: *Response Parameters*. Additionally, any parameters specified in the Merchant User's merch_custom_fields are valid response parameters.

To see enhanced information such as the raw values sent back by the processor, see Section 5.6.1: Get Raw Data.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**trandetail**
ttid	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
last_modified_ts	NS	Timestamp of last transaction modification, YYYY-MM-DD HH:MM:SS +/-ZZZZ
merch_proc	ANS	Name of processing institution used
orig_code	A	code from the original transaction
orig_msoft_code	ANS	msoft_code from the original transaction
orig_phard_code	ANS	phard_code from the original transaction
orig_reqamount	M	amount from the original transaction
orig_verbiage	ANS	verbiage from the original transaction
In addition to the fields above, this report also returns various request and response parameters from the original transaction. See Appendix A: *Request Parameters* and Appendix B: *Response Parameters*.		
In addition to the fields above, this report also returns any custom fields set for the merchant. See Section 5.1.8: Get Merchant Information.		

5.6.6 Get Unsettled Batch Totals

Get a datablock of unsettled batches, optionally limited by date or a provided batch number. Each record will include detailed totals for the batch.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**bt**
batch	O	N, Max 10	Batch number
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.

Report Fields		
Key	**Spec**	**Description**
BatchNum	N, Max 10	Batch number
sub	N, Max 10	Subaccount for batch
status	A	Batch status. Possible values: open - Transactions can be added to batch closed - Transactions cannot be added to batch
totaltransNum	N	Total number of authorizations and returns in the batch
totaltransAmount	M	Aggregate authorization amounts minus aggregate return amounts
totalAuthNum	N	Total number of authorizations
totalAuthAmount	M	Aggregate amount of all authorizations
totalReturnNum	N	Total number of returns
totalReturnAmount	M	Aggregate amount of all returns
totaltransExamount	M	For Restaurant, authorization examounts minus return examounts
totalAuthExamount	M	For Restaurant, aggregate examount of all authorizations
totalReturnExamount	M	For Restaurant, aggregate examount of all returns
NumVisaAuth	N	Total number of Visa authorizations
AmntVisaAuth	M	Aggregate amount of all Visa authorizations
NumVisaReturn	N	Total number of Visa returns
AmntVisaReturn	M	Aggregate amount of all Visa returns
NumVisaDSAuth	N	Total number of Discover authorizations
AmntVisaDSAuth	M	Aggregate amount of all Discover authorizations
NumVisaDSReturn	N	Total number of Discover returns
AmntVisaDSReturn	M	Aggregate amount of all Discover returns
NumMCAuth	N	Total number of Mastercard authorizations
AmntMCAuth	M	Aggregate amount of all Mastercard authorizations
NumMCReturn	N	Total number of Mastercard returns
AmntMCReturn	M	Aggregate amount of all Mastercard returns
NumDiscAuth	N	Total number of Discover authorizations

Report Fields		
Key	**Spec**	**Description**
AmntDiscAuth	M	Aggregate amount of all Discover authorizations
NumDiscReturn	N	Total number of Discover returns
AmntDiscReturn	M	Aggregate amount of all Discover returns
NumCUPAuth	N	Total number of China UnionPay authorizations
AmntCUPAuth	M	Aggregate amount of all China UnionPay authorizations
NumCUPReturn	N	Total number of China UnionPay returns
AmntCUPReturn	M	Aggregate amount of all China UnionPay returns
NumAmexAuth	N	Total number of American Express authorizations
AmntAmexAuth	M	Aggregate amount of all American Express authorizations
NumAmexReturn	N	Total number of American Express returns
AmntAmexReturn	M	Aggregate amount of all American Express returns
NumDinersAuth	N	Total number of Diners Club authorizations
AmntDinersAuth	M	Aggregate amount of all Diners Club authorizations
NumDinersReturn	N	Total number of Diners Club returns
AmntDinersReturn	M	Aggregate amount of all Diners Club returns
NumCBAuth	N	Total number of Carte Blanche authorizations
AmntCBAuth	M	Aggregate amount of all Carte Blanche authorizations
NumCBReturn	N	Total number of Carte Blanche returns
AmntCBReturn	M	Aggregate amount of all Carte Blanche returns
NumJCBAuth	N	Total number of Japan Credit Bureau authorizations
AmntJCBAuth	M	Aggregate amount of all Japan Credit Bureau authorizations
NumJCBReturn	N	Total number of Japan Credit Bureau returns
AmntJCBReturn	M	Aggregate amount of all Japan Credit Bureau returns
NumGIFTAuth	N	Total number of Gift Card authorizations

Report Fields		
Key	**Spec**	**Description**
AmntGIFTAuth	M	Aggregate amount of all Gift Card authorizations
NumGIFTReturn	N	Total number of Gift Card returns
AmntGIFTReturn	M	Aggregate amount of all Gift Card returns
NumOtherAuth	N	Total number of uncategorised authorizations
AmntOtherAuth	M	Aggregate amount of all uncategorised authorizations
NumOtherReturn	N	Total number of uncategorised returns
AmntOtherReturn	M	Aggregate amount of all uncategorised returns
NumDebitAuth	N	Total number of Debit authorizations
AmntDebitAuth	M	Aggregate amount of all Debit authorizations
NumDebitReturn	N	Total number of Debit returns
AmntDebitReturn	M	Aggregate amount of all Debit returns
NumEBTAuth	N	Total number of EBT authorizations
AmntEBTAuth	M	Aggregate amount of all EBT authorizations
NumEBTReturn	N	Total number of EBT returns
AmntEBTReturn	M	Aggregate amount of all EBT returns
NumCheckAuth	N	Total number of check authorizations
AmntCheckAuth	M	Aggregate amount of all check authorizations
NumACHAuth	N	Total number of ACH authorizations
AmntACHAuth	M	Aggregate amount of all ACH authorizations
NumACHReturn	N	Total number of ACH returns
AmntACHReturn	M	Aggregate amount of all ACH returns
NumUnknownAuth		Unused
AmntUnknownAuth		Unused
NumUnknownReturn		Unused
AmntUnknownReturn		Unused

5.6.7 Get Settled Batch Totals

Get a datablock of settled batches, optionally limited by date or a provided batch number. Each record will include detailed totals for the batch. One of batch, bdate/edate, or reversible must be specified.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**pbt**
batch	C	N, Max 10	Batch number
bdate	C	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	C	ANS	Ending date. See Appendix C.8: Date Formats.
reversible	O	A	Reversible status. If not sent, both types will be shown. Possible values: yes - Show only transactions that can be reversed or unsettled no - Show only transactions that cannot be reversed or unsettled both - Show both types

Report Fields		
Key	**Spec**	**Description**
BatchNum	N, Max 10	Batch number
sub	N, Max 10	Subaccount for batch
timestamp	ANS	Settlement timestamp, YYYY-MM-DD HH:MM:SS +/-ZZZZ
totaltransNum	N	Total number of authorizations and returns in the batch
totaltransAmount	M	Aggregate authorization amounts minus aggregate return amounts
totalAuthNum	N	Total number of authorizations
totalAuthAmount	M	Aggregate amount of all authorizations
totalReturnNum	N	Total number of returns
totalReturnAmount	M	Aggregate amount of all returns
totaltransExamount	M	For Restaurant, authorization examounts minus return examounts
totalAuthExamount	M	For Restaurant, aggregate examount of all authorizations
totalReturnExamount	M	For Restaurant, aggregate examount of all returns

Report Fields		
Key	**Spec**	**Description**
NumVisaAuth	N	Total number of Visa authorizations
AmntVisaAuth	M	Aggregate amount of all Visa authorizations
NumVisaReturn	N	Total number of Visa returns
AmntVisaReturn	M	Aggregate amount of all Visa returns
NumVisaDSAuth	N	Total number of Discover authorizations
AmntVisaDSAuth	M	Aggregate amount of all Discover authorizations
NumVisaDSReturn	N	Total number of Discover returns
AmntVisaDSReturn	M	Aggregate amount of all Discover returns
NumMCAuth	N	Total number of Mastercard authorizations
AmntMCAuth	M	Aggregate amount of all Mastercard authorizations
NumMCReturn	N	Total number of Mastercard returns
AmntMCReturn	M	Aggregate amount of all Mastercard returns
NumDiscAuth	N	Total number of Discover authorizations
AmntDiscAuth	M	Aggregate amount of all Discover authorizations
NumDiscReturn	N	Total number of Discover returns
AmntDiscReturn	M	Aggregate amount of all Discover returns
NumCUPAuth	N	Total number of China UnionPay authorizations
AmntCUPAuth	M	Aggregate amount of all China UnionPay authorizations
NumCUPReturn	N	Total number of China UnionPay returns
AmntCUPReturn	M	Aggregate amount of all China UnionPay returns
NumAmexAuth	N	Total number of American Express authorizations
AmntAmexAuth	M	Aggregate amount of all American Express authorizations
NumAmexReturn	N	Total number of American Express returns
AmntAmexReturn	M	Aggregate amount of all American Express returns
NumDinersAuth	N	Total number of Diners Club authorizations
AmntDinersAuth	M	Aggregate amount of all Diners Club authorizations
NumDinersReturn	N	Total number of Diners Club returns

Report Fields		
Key	**Spec**	**Description**
AmntDinersReturn	M	Aggregate amount of all Diners Club returns
NumCBAuth	N	Total number of Carte Blanche authorizations
AmntCBAuth	M	Aggregate amount of all Carte Blanche authorizations
NumCBReturn	N	Total number of Carte Blanche returns
AmntCBReturn	M	Aggregate amount of all Carte Blanche returns
NumJCBAuth	N	Total number of Japan Credit Bureau authorizations
AmntJCBAuth	M	Aggregate amount of all Japan Credit Bureau authorizations
NumJCBReturn	N	Total number of Japan Credit Bureau returns
AmntJCBReturn	M	Aggregate amount of all Japan Credit Bureau returns
NumGIFTAuth	N	Total number of Gift Card authorizations
AmntGIFTAuth	M	Aggregate amount of all Gift Card authorizations
NumGIFTReturn	N	Total number of Gift Card returns
AmntGIFTReturn	M	Aggregate amount of all Gift Card returns
NumOtherAuth	N	Total number of uncategorised authorizations
AmntOtherAuth	M	Aggregate amount of all uncategorised authorizations
NumOtherReturn	N	Total number of uncategorised returns
AmntOtherReturn	M	Aggregate amount of all uncategorised returns
NumDebitAuth	N	Total number of Debit authorizations
AmntDebitAuth	M	Aggregate amount of all Debit authorizations
NumDebitReturn	N	Total number of Debit returns
AmntDebitReturn	M	Aggregate amount of all Debit returns
NumEBTAuth	N	Total number of EBT authorizations
AmntEBTAuth	M	Aggregate amount of all EBT authorizations
NumEBTReturn	N	Total number of EBT returns
AmntEBTReturn	M	Aggregate amount of all EBT returns
NumCheckAuth	N	Total number of check authorizations
AmntCheckAuth	M	Aggregate amount of all check authorizations
NumACHAuth	N	Total number of ACH authorizations
AmntACHAuth	M	Aggregate amount of all ACH authorizations

Report Fields		
Key	**Spec**	**Description**
`NumACHReturn`	N	Total number of ACH returns
`AmntACHReturn`	M	Aggregate amount of all ACH returns
`NumUnknownAuth`		Unused
`AmntUnknownAuth`		Unused
`NumUnknownReturn`		Unused
`AmntUnknownReturn`		Unused

5.7 History Maintenance

The requests in this section allow Main Merchant Users to perform house-cleaning duties on the system.

These are the various transactions and records that will be cleared for each action:

admin	Uncaptured transactions	Voided transactions	Failed transactions	Settled transactions	Settlement record	Sensitive data
cth		X		X	X	
cut	X					
cfh			X			
securetrans						X

5.7.1 Clear Transaction History

Clear voided and settled transactions and settlement records from the history.

Voided transactions can be seen with the gut and gl reports using type=void.

Settled transactions can be seen with the gl report.

Settlement records can be seen with the pbt report.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**cth**
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.
batch	O	N, Max 10	Batch number
keeptotals	O	A	Boolean flag indicating whether or not to keep settlement records. Defaults to no.

Response Parameters

Key	Spec	Description
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.7.2 Clear Uncaptured Transactions

Clear uncaptured transactions from the history.

This is used to remove stale transactions that would not be funded if settled. To prevent accidental misuse, this request can only operate on transactions that are at least 30 days old. Typically, settlement of a transaction over 30 days old will result in a rejection of funding. If an old transaction needs to be collected, a new charge should be issued.

This request only works on *uncaptured* transactions, meaning ones that have been authorized but not added to a batch. This can be from either a **`sale`** sent with `capture=no` or a **`preauthorization`**.

Uncaptured transactions can be seen with the `gut` report using `capture=no`.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters

Key	Req	Spec	Description
action	Y		**admin**
admin	Y		**cut**
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.7.3 Clear Failed History

Clear failed transactions from the history.

Failed transactions can be seen with the gft report.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**cfh**
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.7.4 Secure Transactions

Clear sensitive data from the transaction history.

This will remove PCI-protected data like account information from the transaction history while keeping the record intact. An integrator might use this request to further secure their systems by removing sensitive data that is longer relevant for business operations.

Note: Transactions that have had their sensitive data scrubbed will no longer be eligible for future transactions run using their ttid, including refunds, duplicate sales, and settlements.

Access level: Section 2.4.2: Merchant User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**securetrans**
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route
batch	O	N, Max 10	Batch number
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.8 Level III

Level III processing (aka Enhanced Processing) is a feature specifically designed for processing commercial cards and government purchasing cards within a business-to-business and/or business-to-government environment. In order to process Level III transactions successfully, you must be able to supply detailed line item information to complete the transaction. The required line item details are basically the same as the details a customer might see on an invoice, such as products purchased, SKU numbers, descriptions, quantities, etc.

Both Visa and Mastercard are currently supported for Level III transactions.

Performing Level III transactions is a two-step process in Monetra:

1. Run a regular authorization (either a `sale` or `preauthorization`), including some additional parameters, as outlined below.
2. Add Level III line items to the authorization. It should be noted that line items can be added to and deleted from only unsettled transactions.

Note: You must be specifically licensed to support Level III features and use a processor that supports the same.

When performing the initial Level III authorization, these parameters are required or conditional based on other factors:

Key	Req	Spec	Description
amount	Y	M	Total amount of transaction, including all subtotals and supplementary amounts. This is always the amount to be settled.
ordernum	Y	AN, Max 25	Merchant order number or hotel folio
tax	Y	M	Amount of total that is tax applied to the order, or nt if the transaction is non-taxable/tax-exempt
custref	Y	AN, Max 17	Customer reference number, or PO number. Defaults to value of ordernum.
discountamount	C	M	Amount of discount applied to the order as a whole. This includes all discounts for each line item. The sum of all line item discounts must be less than or equal to this value.
freightamount	C	M	Amount of freight/shipping on the order. If shipping was charged on the order, this field must be sent along with shipzip. Freight is never part of the individual line items.
dutyamount	C	M	Total amount of duty (fee associated with import of goods) on order
shipzip	C	AN	Zip code where product is being shipped. Defaults to merchant's billing zip code. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
shipcountry	C	AN, 3	Country where product is being shipped, formatted as ISO 3166-1 numeric or alpha-3

5.8.1 Add Line Item

Add a new Level III line item.

Each line item must have certain parameters, as seen in the table below, while others are optional. Additionally, some parameters have maximum values. Unless otherwise stated, all amounts have a maximum decimal precision of 5 digits. Typically, there is a maximum of 97 line items per transaction. If more are needed, the transaction should be broken up into smaller transactions.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**l3add**
ttid	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
commoditycode	Y	ANS, 8-12	International description code of the individual good or service being supplied The acquiring bank or processor should provide the merchant an updated listing of currently defined codes. Please note that this is different from the 4-character summary commodity code. Codes can be found online at http://www.unspsc.org. The code itself will not be validated; only the format is validated.
description	Y	ANS, Max 26	Merchant-defined description of the item or service being sold
productcode	Y	ANS, Max 12	Merchant-defined code for the item being purchased, such as a UPC #, SKU #, or Item #
qty	Y	N	Quantity of the item being purchased (Maximum value of 99,999 units)
unit	Y	ANS, Max 12	Merchant-defined unit of measure
unitprice	Y	M	Item price per unit, without duty, freight, discount, or tax (Maximum value of 9,999,999.99)
discountamount	O	M	Discount amount per line item (Maximum value of 9,999,999.99, no more than 2 decimals)
discountrate	O	M	Discount rate per line item (Maximum value of 99.99)
amount	Y	M	Line item total amount after all taxes and discounts have been applied (but not inclusive of duty or freight) (Maximum value of 9,999,999.99, no more than 2 decimals)

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
l3id	N, Max 19	ID associated with the Level III line item

5.8.2 Delete Line Item

Delete an existing Level III line item.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**l3del**
l3id	Y	N, Max 19	ID associated with the Level III line item

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.8.3 List Line Items

Get a datablock of Level III line items for a transaction, optionally limited to a single provided line item.

Either the ttid of the transaction or the l3id of a line item must be provided.

Access level: Section 2.4.2: Merchant User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**l3list**
ttid	C	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
l3id	C	N, Max 19	ID associated with the Level III line item

Report Fields		
Key	**Spec**	**Description**
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
l3id	N, Max 19	ID associated with the Level III line item
commoditycode	ANS, 8-12	International description code of the individual good or service being supplied The acquiring bank or processor should provide the merchant an updated listing of currently defined codes. Please note that this is different from the 4-character summary commodity code. Codes can be found online at http://www.unspsc.org. The code itself will not be validated; only the format is validated.
description	ANS, Max 26	Merchant-defined description of the item or service being sold
productcode	ANS, Max 12	Merchant-defined code for the item being purchased, such as a UPC #, SKU #, or Item #

Report Fields		
Key	**Spec**	**Description**
qty	N	Quantity of the item being purchased (Maximum value of 99,999 units)
unit	ANS, Max 12	Merchant-defined unit of measure
unitprice	M	Item price per unit, without duty, freight, discount, or tax (Maximum value of 9,999,999.99)
discountamount	M	Discount amount per line item (Maximum value of 9,999,999.99, no more than 2 decimals)
discountrate	M	Discount rate per line item (Maximum value of 99.99)
amount	M	Line item total amount after all taxes and discounts have been applied (but not inclusive of duty or freight) (Maximum value of 9,999,999.99, no more than 2 decimals)

5.9 Image Storage

5.9.1 Add Image

Link an image to a transaction and store it in Monetra.

If an image of the provided type is already stored with the transaction, this will overwrite it.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**imageadd**
image	Y	ANS	Base64-encoded image data, formatted as TIFF, PNG, or PBM, less than 64k in size
image_type	Y	A	Type of image captured. Possible values: signature check
ttid	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.9.2 Delete Image

Delete an existing image stored with a transaction.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**imagedel**
image_type	Y	A	Type of image captured. Possible values: signature check
ttid	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.9.3 List Images

Get a datablock of images stored in Monetra, optionally limited to specific data.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**getimages**
image_format	O	A	Image format. Possible values: tiff (default) png pbm
ttid	O	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
batch	O	N, Max 10	Batch number
status	O	A	Status of transaction. Possible values: unsettled settled
bdate	O	ANS	Beginning date. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date. See Appendix C.8: Date Formats.

Report Fields		
Key	**Spec**	**Description**
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
ts	NS	Timestamp of when image was added, YYYY-MM-DD HH:MM:SS +/-ZZZZ
status	A	Status of transaction. Possible values: unsettled settled
ptrannum	N, Max 19	Similar to ordernum, but numeric for efficient indexing in the database. It is recommended to use ordernum instead.
batch	N, Max 10	Batch number
image_type	A	Type of image captured. Possible values: signature check
image	ANS	Base64-encoded image data, formatted as TIFF, PNG, or PBM, less than 64k in size

5.10 Automated Merchant Task Management

Monetra has a built-in task scheduler known by the common Unix name of "Cron". It is capable of scheduling various tasks--such as history purges, settlements, and token management--on a periodic basis.

The Cron subsystem will email the results of each task to the specified addresses. For Merchant User tasks, the Cron email system will use the email as specified in the merchant configuration. Monetra must be properly configured to send emails either via a local instance of `sendmail` or a valid `SMTP` server location. For more information on this, see the Secure Install Guide here [https://www.monetra.com/documentation].

For Monetra users, global settings pertaining to the Cron subsystem can be set in the `main.conf` file in the `Misc` section.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**cron**
cron	Y	A	The Cron function to perform. See Section 5.10.1: Cron Functions.
cron_date	C	ANS	Frequency to run the task. See Section 5.10.2: Cron Date Format.
cron_task	C	A	Cron task to schedule. See Section 5.10.3: Cron Tasks.
cron_data	C	N	Data specific to a Cron task. See Section 5.10.4: Cron Data.
cronid	C	N, Max 20	Only used for remove and run_task functions, unique cron identifier, as returned from audit

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

5.10.1 Cron Functions

The following functions may be performed through the Cron subsystem:

cron	Description
add	Add the task to the Cron schedule
list	List all scheduled tasks
remove	Remove the task from the Cron schedule
run_task	Run the task immediately. This will not schedule the task. It will only be run one time.

5.10.2 Cron Date Format

The date format is very flexible to allow a wide range of dates. You can specify one or more days of the week to run the task or certain days of the month, but you can only specify one time per task. The format is:

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

Format	Description
<time>	Represented as HHMM, assuming a 24hr clock. (e.g. 1430 is 2:30pm)
<day/date>	The standard, three-letter abbreviation for the day of the week, or 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
0200|mon;thu	2:00am every Monday and Thursday
1200|1;15	12:00pm on the 1st and 15th of each month
2200|*	10:00pm everyday

5.10.3 Cron Tasks

Main Merchant Users may specify one of the following tasks:

cron_task	Description
settle	Settles all unsettled batches, as seen via Section 5.6.6: Get Unsettled Batch Totals. Same as Section 6.5.1: Settle Batch.
securehist	Clears sensitive data from the transaction history. Same as Section 5.7.4: Secure Transactions.
purgehist	Clear failed, voided, and settled transactions from the history. Same as Section 5.7.3: Clear Failed History and Section 5.7.1: Clear Transaction History together.
purgeexpired	Purges expired cards/accounts within the token system. Same as Section 5.2.8: Delete Expired Tokens.
purgerecurhist	Purges the history of recurring and installment payment attempts within the recurring billing and storage system. Same as Section 5.2.7: Clear Token History.
purgeuncaptured	Purges uncaptured transactions from the history. Same as Section 5.7.2: Clear Uncaptured Transactions.

Note: Depending on the Cron task requested, you might receive a different response. For example, cron=list will return a comma-separated list of scheduled tasks with the these headers/fields: cronid,cron_task,cron_date,cron_data,last_ts,next_ts

5.10.4 Cron Data

Some of the tasks above require additional data to perform their duty. The table below summarizes any necessary information.

Note: The number of days/months must be positive integers.

cron_data	Description
securehist	Number of days to keep
purgehist	Number of days to keep
purgeexpired	Number of months to keep (optional)
purgerecurhist	Number of days to keep
purgeuncaptured	Number of days to keep

5.11 Transaction Export/Import

Note: While TranSafe can be part of this process, it is really meant to be carried out between multiple instances of Monetra.

Exporting/importing transaction data is used to facilitate the concept of an "overhead authorization system", meaning one system authorizes a transaction that a different store will settle. This section details the steps needed to transfer the transaction data from one Monetra instance to another.

Consider this example: A company has one website and multiple physical stores. There is a Monetra instance running for the website's Ecommerce shopping cart as well as one for each of the stores. A customer can place an order through the website that will be fulfilled by one of the stores. The Ecommerce instance of Monetra will handle the actual authorization, but the fulfilling store needs the transaction details for bookkeeping purposes. In this example, the Ecommerce Monetra instance would carry out the sale and then encrypt and export the unsettled transaction data. It would then send that data to the store fulfilling the order, which would import and decrypt the data. At this point, the physical store would have the record of the sale as if it had authorized the transaction itself. If the transaction was captured, it will be added to the open batch and settled at this store.

Unless the Merchant User's config allows PAN access globally or does not have the `obscure` flag set, the transaction data must be encrypted. Continuing with the example above, the physical stores would each create an RSA public/private key pair. This is done either internally by provisioning a CardShield RSA key pair using `keytype=rsa` or externally by creating an RSA key pair. If done internally, then the private key will be securely stored in Monetra. If done externally, then the path to the private key will need to be set in `txn_import_key` in `main.conf`. In both cases, the public key will be sent to the Ecommerce instance, which will then add it to the associated Merchant User in either the initial Add User request or later with an Edit User request, using the parameter `txn_export_key`, as detailed in Section 4.3.1: Add Merchant User.

5.11.1 Export Transaction

Export the record of a specified transaction.

To do this automatically in a transaction request, see the `txnexport` parameter.

Note: Only unsettled transactions are eligible for export.

`Access level:`	Section 2.4.2: Merchant User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
action	Y		**txnexport**
ttid	Y	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction
txnexport	O	A	If set to encrypted, encrypt exported data packet

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
txnexportdata	ANS	Base64-encoded export data packet, for use with a subsequent Transaction Import

5.11.2 Import Transaction

Import a transaction from a previously-exported packet.

This will generate a new ttid that is specific to the destination account.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**admin**
admin	Y		**txnimport**
txndata	Y	ANS	Base64-encoded export data packet, as obtained from a Transaction Export

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	AS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
ttid	N, Max 19	Unique ID referencing a transaction within Monetra, as returned with the original transaction

6 Merchant Subuser Actions

The following sections in this chapter detail the actions that Merchant Subusers with sufficient privileges are allowed to perform.

Please review the required permission level to send Merchant User actions. See Section 2.4.2: Merchant User Authentication.

6.1 Credit Card and Debit Card Transactions

The actions in this section apply to credit cards and debit cards.

6.1.1 Sale

Authorize a hold on funds and add the transaction to a batch for later settlement.

The funds will be delivered to the merchant after a successful settlement.

The transaction amount is assumed to be the final transaction amount that will be settled. If the transaction amount is subject to change, a preauth request should be used instead.

To use account data from a previous transaction, pass the previous transaction's `ttid`. Any previous transaction can be used, as long as it hasn't been secured or purged.

Note: If a sale needs to be canceled before it has been settled, then a **`reversal`** must be issued. This will both remove the hold on the authorized funds and remove the transaction from the batch.

`Access level:`	Section 2.4.2: Merchant User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`sale`**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Verification Data	O		Various parameters. See Appendix A.3: Verification Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Token Data	O		Various parameters. See Appendix A.6: Token Information.
COF Data	O		Various parameters. See Appendix A.7: COF/Recurring Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.

Request Parameters			
Key	**Req**	**Spec**	**Description**
Shipping Data	O		Various parameters. See Appendix A.10: Shipping Information.
Ecommerce Data	O		Various parameters. See Appendix A.11: Ecommerce Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.
`ttid`	C	N, Max 19	Unique ID returned with every transaction. Required if referencing a previous transcation to reuse account data.
`capture`	O	A	Boolean flag indicating whether or not the approved transaction should be added to the batch. Defaults to `yes`. If `no`, the transaction is functionally equivalent to a **`preauth`**.

Response Parameters		
Key	Spec	Description
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
authamount	M	Amount of funds actually authorized. This is only present when the authorized amount is less than the requested amount, as can happen with insufficient funds. If all of the requested amount is authorized, then this will not appear in the response.
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.2 Reversal

Remove the hold on funds from a previous authorization.

Some card brands allow partial reversals, where only part of the hold is removed. This is beneficial in some industries such as Ecommerce where an order might be changed before the transaction is completed.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	Req	Spec	Description
action	Y		**reversal**
ttid	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
reversal_reason	O	A	Reason for issuing the reversal/void. Possible values: clerkvoid - Clerk requested to void the transaction (default) customercancel - Customer canceled transaction after approval (only if

Request Parameters			
Key	**Req**	**Spec**	**Description**
			customer hits cancel; use clerkvoid if customer instructs clerk to cancel) deliveryfailure - Response could not be delivered to the POS devicefailure - Could not communicate with EMV terminal cardremoved - Card was removed prior to processor response carddecline - Chip card declined the transaction after it was approved online macfailure - Interac MAC verification failure fraud - Fraud suspected (default for fraudautodeny transactions)
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.3 Return

Refund a prior purchase.

If the transaction being referenced is still unsettled, the return amount must be different than the original authorization amount. If the amounts are the same, a reversal should be issued instead.

For credit cards, instead of sending the sensitive account info, a prior transaction's `ttid` can be referenced (if the transaction hasn't been secured or purged).

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`return`**
`ttid`	C	N, Max 19	Unique ID returned with every transaction. Required if referencing a previous transcation to reuse account data.
Account Data	C		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.4 Preauthorization

Authorize a hold on funds *without* adding the transaction to a batch.

A **preauth** is essentially the first part of a **sale**. A regular **sale** authorizes a hold on funds and adds the transaction to a batch, while a **preauth** authorizes the hold but does not add the transaction to a batch. This means that a **preauth** cannot be settled by itself. To add a **preauth** to a batch, thereby making it eligible for settlement, you need to issue a **preauthcomplete**.

The transaction amount can change between the **preauth** and **preauthcomplete**. You should use this if the final amount is not known at the time of authorization, e.g. adding a tip.

Some industries, such as Ecommerce, necessitate the use of a **preauth**. When selling physical goods, the final charge (settle) cannot take place until the goods are shipped. If there is a short delay between taking payment and shipping, a **preauth** can be used to delay taking the funds from a customer's account. You might need to work with your merchant account provider to determine whether a **sale** or **preauth** is more appropriate for your business needs.

To use account data from a previous transaction, pass the previous transaction's ttid. Any previous transaction can be used, as long as it hasn't been secured or purged.

Access level: Section 2.4.2: Merchant User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`preauth`
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Verification Data	O		Various parameters. See Appendix A.3: Verification Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Token Data	O		Various parameters. See Appendix A.6: Token Information.
COF Data	O		Various parameters. See Appendix A.7: COF/Recurring Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Shipping Data	O		Various parameters. See Appendix A.10: Shipping Information.
Ecommerce Data	O		Various parameters. See Appendix A.11: Ecommerce Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.
`ttid`	C	N, Max 19	Unique ID returned with every transaction. Required if referencing a previous transcation to reuse account data.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
authamount	M	Amount of funds actually authorized. This is only present when the authorized amount is less than the requested amount, as can happen with insufficient funds. If all of the requested amount is authorized, then this will not appear in the response.
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.5 Preauthorization Complete

Finalize a **preauth**.

This adds the transaction to a batch so that it can be settled and the funds can be transferred.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**preauthcomplete**
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.6 Force

Add an authorized transaction to a batch.

To add a transaction that was authorized outside of Monetra, you must send the full transaction data, including the account data, amount, and approval code.

`Access level:` Section 2.4.2: Merchant User Authentication
`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters

Key	Req	Spec	Description
`action`	Y		**`force`**
`apprcode`	Y	AN	Approval code (aka authorization number) that the processing institution returned for the authorization, whether from a phone auth or another device. Not necessary if a previous transaction is referenced by `ttid`.
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Verification Data	O		Various parameters. See Appendix A.3: Verification Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
Token Data	O		Various parameters. See Appendix A.6: Token Information.
COF Data	O		Various parameters. See Appendix A.7: COF/Recurring Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Shipping Data	O		Various parameters. See Appendix A.10: Shipping Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transcription result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.7 Capture

Add an uncaptured **`sale`** to a batch.

If a **`sale`** was sent with `capture=no` and approved, then a hold has been placed on funds, but the transaction was not added to a batch and is not eligible for settlement. This will add the uncaptured transaction to a batch and make it eligible for settlement.

This is functionally equivalent to a **`preauthcomplete`**, except that the original transaction is a **`sale`** instead of a **`preauth`**.

There are a number of reasons to use this flow over the **`preauth`**/**`preauthcomplete`** flow, including the necessity for some MCCs to authorize and settle for the same amount. You should work with your merchant account provider to determine the appropriate flow for your business needs.

`Access level:`	Section 2.4.2: Merchant User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**capture**
ttid	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.8 AVS Only

Check a card's legitimacy.

There are three main uses for this action:

1. Verify the card is valid and open
2. Verify AVS and CV data
3. Tokenize the card

This does not verify availability of funds, and it does not place a hold on any funds.

To use account data from a previous transaction, pass the previous transaction's ttid. Any previous transaction can be used, as long as it hasn't been secured or purged.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**avsonly**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Verification Data	Y		Various parameters. See Appendix A.3: Verification Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Token Data	O		Various parameters. See Appendix A.6: Token Information.
COF Data	O		Various parameters. See Appendix A.7: COF/Recurring Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Ecommerce Data	O		Various parameters. See Appendix A.11: Ecommerce Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
ttid	C	N, Max 19	Unique ID returned with every transaction. Required if referencing a previous transcation to reuse account data.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.9 Card Type

Get information on a card without running a financial transaction.

This will perform a real-time card type look-up against the system's current BIN table. Unlike an `avsonly` request, this is a local card check that stays in Monetra; no information will be sent out to the processing institution.

This should be used when you need to know what type of card is being presented in advance of a financial request. It is particularly useful for integrated systems that do not maintain their own internal BIN table.

Note: When using TranSafe or running a Monetra instance with a Global BIN file configured, extended metadata might be returned. Which fields are returned depends on relevance and presence in the Global BIN file.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`cardtype`**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Token Data	O		Various parameters. See Appendix A.6: Token Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
`ttid`	C	N, Max 19	Unique ID returned with every transaction. Required if referencing a previous transcation to reuse account data.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.10 Adjust

Modify transaction data.

Sometimes, not all parameters are required or known at the time of authorization, and some might change afterwards. Using this action, you can add, modify, and remove transaction parameters after the initial authorization.

This can also edit fields that affect interchange rates (e.g `tax`, `rate`, and `bdate`/`edate`). If an interchange parameter is not known or finalized, it can be added or updated.

When dealing with changes to amounts, it is typically better to use a **`preauth`** and **`preauthcomplete`** instead of adjusting a sale. WorldPay's 610 platform is a special case where it is better to use **`sale`** and **`adjust`**.

Note: This will only affect transactions that are currently unsettled. It should be used with captured transactions. It should not be used with a **`preauth`** that has not had a corresponding **`preauthcomplete`**. In that case, the transaction should be adjusted when issuing the **`preauthcomplete`**.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**adjust**
ttid	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.11 Incremental (Lodging)

Increase an authorization's amount.

The amount sent with this action should be the amount to be settled, *not* the amount to increment.

Note: This is for transactions run with Lodging MCCs only.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**incremental**
ttid	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.1.12 Void

Delete a transaction from within Monetra.

In the vast majority of cases, you should use a **reversal** instead of a **void**.

A **void** removes the transaction from within Monetra, but it usually does not go online and get sent to a processor. This means that the hold on the funds for which the transaction was originally authorized will not be removed, and the customer will not be able to access those funds until the hold expires.

When a transaction has been removed from Monetra, you cannot see/modify/interact with it again.

The main use for a **void** is to ensure that the transaction will not settle if a **reversal** failed. Because a **void** is typically offline and never leaves Monetra, there is still a chance of success, with the right processor. A **reversal** can fail for legitimate reasons, such as the transaction already being settled, in which case a **return** should be issued. Another case is when the transaction has already been reversed. If a **reversal** fails, you should evaluate the response to determine if a **void** is actually necessary as a follow-up.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**void**
ttid	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
reversal_reason	O	A	Reason for issuing the reversal/void. Possible values: clerkvoid - Clerk requested to void the transaction (default) customercancel - Customer canceled transaction after approval (only if customer hits cancel; use clerkvoid if customer instructs clerk to cancel) deliveryfailure - Response could not be delivered to the POS devicefailure - Could not communicate with EMV terminal cardremoved - Card was removed prior to processor response carddecline - Chip card declined the transaction after it was approved online macfailure - Interac MAC verification failure fraud - Fraud suspected (default for fraudautodeny transactions)
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.

Request Parameters			
Key	**Req**	**Spec**	**Description**
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.2 EBT Transactions

The actions in this section apply to Electronic Benefit Transfer (EBT) cards. Some are for Food Stamps, and some are for Cash Benefits cards.

6.2.1 Food Stamps Transactions

6.2.1.1 Sale

Debit funds from a Food Stamps account.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`ebtfssale`
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.2.1.2 Return

Refund a previously settled amount to a Food Stamps account.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`ebtfsreturn`**
`ttid`	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.2.1.3 Voucher Sale

Add a Food Stamps sale to Monetra that was authorized through another system (e.g. over the phone).

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**ebtfsvoucher**
voucherapproval	Y	AN, Max 6	Voucher approval number from issuer, not required for all processors
voucherserial	Y	AN, Max 15	Voucher serial number from issuer, not required for all processors
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.2.1.4 Balance Inquiry

Get the balance remaining in a Food Stamps account.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**ebtfsbalance**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.2.2 EBT Cash Benefits Transactions

6.2.2.1 Sale

Debit funds from a Cash Benefits account.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**ebtcbsale**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.2.2.2 Cash Withdrawal

Debit funds from a Cash Benefits account, with the full amount given as cash to the customer.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**ebtcbcash**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.2.2.3 Balance Inquiry

Get the balance remaining in a Cash Benefits account.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**ebtcbbalance**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
PIN Data	O		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3 Gift Card Transactions

The actions in this section apply to gift cards.

6.3.1 Activate

Activate a gift card.

Typically, this is used for a gift card that has a preset amount. When the card is activated, it comes with that amount loaded already. This means you don't need to pass an `amount` with this request. To activate a card and load it with a specific amount, use **`issue`** instead.

`Access level:`	Section 2.4.2: Merchant User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		**`activate`**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	O		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	C		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3.2 Issue

Activate and load a gift card.

Typically, this is used for a gift card that does not come with a value pre-loaded. This activates the gift card and loads it for the amount specified. To activate a card that has a preset amount, use **activate** instead.

Access level: Section 2.4.2: Merchant User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`issue`
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	C		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3.3 Balance Inquiry

Get the balance remaining on the gift card.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`balanceinq`
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
PIN Data	C		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
`code`	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
`msoft_code`	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
`phard_code`	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
`verbiage`	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3.4 Redemption

Debit funds from a gift card.

This decreases the amount of funds available on the card.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		redemption
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	C		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3.5 Tip

Add a tip to a gift card transaction.

This is meant to be used after the original transaction, when the card is no longer present. This decreases the amount of funds available on the card.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters

Key	Req	Spec	Description
action	Y		**tip**
ttid	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters

Key	Spec	Description
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3.6 Cash Out

Debit all remaining funds from a gift card.

This is used to convert a card's balance into cash. After this, the card will have a balance of $0.00. Some U.S. states mandate support for this action.

Note: Not all gift card processors support this action.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters

Key	Req	Spec	Description
action	Y		cashout
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	C		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters

Key	Spec	Description
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3.7 Reload

Load money onto a gift card.

This increases the amount of funds available on the card.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**reload**
Account Data	Y		Various parameters. See Appendix A.1: Account Information.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	C		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.3.8 Merch Return

Refund a purchase.

This increases the amount of funds available on the card.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters

Key	Req	Spec	Description
action	Y		**merchreturn**
ttid	C	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
Account Data	C		Various parameters. See Appendix A.1: Account Information.
Monetary Data	C		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
PIN Data	C		Various parameters. See Appendix A.5: PIN Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters

Key	Spec	Description
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display

See Appendix B: *Response Parameters* for more possible response parameters.

6.4 Check Transactions

Check transactions are processed through different networks than other card types are. While Credit transactions place holds on available funds and Debit transactions immediately transfer money out of the cardholder's bank account, Check transactions neither place holds nor transfer money right away, and they also don't check if the requested amount is available in the account.

To accept a physical check as a form of payment, you must first capture the check data. While the data can be entered manually, some processors require the use of a MICR reader. When you have the check data, you then convert it into an electronic check before processing it through the bank-to-bank network (ACH). Monetra currently supports four different types of conversion (Section 6.4.2: Convert Check - Section 6.4.5: Convert Check with Override). They all have the same conversion functionality, but some offer different features on top of that. If the conversion is approved, then the transaction has been authorized and the sale approved.

Check and ACH as account types are similar in function, but they differ in where the account information comes from and what fields it has. Check data comes from a physical check that is converted into an electronic check and is tied to that specific check, while ACH data represents the account directly. Check data can be entered manually or read using a MICR reader; ACH data is always entered manually. The actions in this section apply to checks only.

After a physical check is converted, the transaction is processed over the ACH network, which means that the transaction now follows NACHA rules instead of check rules, and a one-time debit is authorized against the account.

Note: Not all of the actions below are supported on all of the Check/ACH processors. See Section 6.4.8.1: Processor Support for a complete mapping.

6.4.1 Verify Check

Verify a check.

This does not actually verify that the check's account exists or is in good standing. While some processors differ in their offerings, this typically performs a fraud test by checking if the account owner is known to write bad checks.

`Access level:`	Section 2.4.2: Merchant User Authentication
`Table legend:`	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**checkverify**
Check Data	Y		Multiple parameters. See Appendix A.1.7: Physical Check.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.4.2 Convert Check

Convert a check.

This is a simple conversion with no additional features.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`checkconvonly`
Check Data	Y		Multiple parameters. See Appendix A.1.7: Physical Check.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
auth	ANS	Authorization code
batch	N, Max 10	Assigned batch number
stan	N	Corresponds to NCN Transaction Number from processor
item	N, Max 10	Assigned item number
bouncefee	M	Amount charged for a returned check
printdata	ANS	Data returned that must be printed on the customer's receipt
See Appendix B: *Response Parameters* for more possible response parameters.		

6.4.3 Convert Check with Verification

Convert a check with verification.

Note: The level and type of verification depends on the processor.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`checkconvvrfy`
Check Data	Y		Multiple parameters. See Appendix A.1.7: Physical Check.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
auth	ANS	Authorization code
batch	N, Max 10	Assigned batch number
stan	N	Corresponds to NCN Transaction Number from processor
bouncefee	M	Amount charged for a returned check
printdata	ANS	Data returned that must be printed on the customer's receipt
See Appendix B: *Response Parameters* for more possible response parameters.		

6.4.4 Convert Check with Guarantee

Convert a check with a guarantee on funds.

Note: The level and type of guarantee depends on the processor.

`Access level:` Section 2.4.2: Merchant User Authentication

`Table legend:` Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`checkconvguar`
Check Data	Y		Multiple parameters. See Appendix A.1.7: Physical Check.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
auth	ANS	Authorization code
batch	N, Max 10	Batch number
stan	N	Corresponds to NCN Transaction Number from processor
bouncefee	M	Amount charged for a returned check
printdata	ANS	Data returned that must be printed on the customer's receipt
See Appendix B: *Response Parameters* for more possible response parameters.		

6.4.5 Convert Check with Override

Convert a check with override.

This overrides a previous failure.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
`action`	Y		`checkconvover`
Check Data	Y		Multiple parameters. See Appendix A.1.7: Physical Check.
Monetary Data	Y		Various parameters. See Appendix A.2: Monetary Information.
Order Data	O		Various parameters. See Appendix A.4: Order Information.
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Receipt Data	O		Various parameters. See Appendix A.9: Receipt Information.
Healthcare Data	O		Various parameters. See Appendix A.12: Healthcare Information.
Lodging Data	O		Various parameters. See Appendix A.13: Lodging Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.
Level III Data	O		Various parameters. See Appendix A.16: Level III Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
auth	ANS	Authorization code
batch	N, Max 10	Batch number
stan	N	Corresponds to NCN Transaction Number from processor
bouncefee	M	Amount charged for a returned check
printdata	ANS	Data returned that must be printed on the customer's receipt
See Appendix B: *Response Parameters* for more possible response parameters.		

6.4.6 Upload Check Image

Upload an image of a check.

Like verifying a check, this is not a financial transaction (i.e. it does not move money). In most instances, this is used to research a check that has been flagged for review.

To retrieve the images associated with a transaction, see Section 5.9.3: List Images.

Note: The initial upload stores the image in Monetra, which will send the image to the processing institution at settlement.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**checkimageupload**
ttid	Y	N, Max 19	Unique ID referencing the conversion transaction to which this check image belongs
image	Y	ANS	Base64-encoded image data, formatted as TIFF, PNG, or PBM, less than 64k in size

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
batch	N, Max 10	Batch number
See Appendix B: *Response Parameters* for more possible response parameters.		

6.4.7 Void

Delete a check conversion from within Monetra, before it goes online for settlement.

Access level: Section 2.4.2: Merchant User Authentication

Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**void**
ttid	Y	N, Max 19	Unique ID referencing the transaction to modify, as returned with the original transaction
Processing Data	O		Various parameters. See Appendix A.8: Processing Information.
Merchant Data	O		Various parameters. See Appendix A.14: Merchant Information.
Lane Data	O		Various parameters. See Appendix A.15: Lane Information.

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.4.8 Processor-Specific Information

This section details information about certain processors that support Check transactions.

6.4.8.1 Processor Support

These are the processing institutions that support the different Check transactions. For more information, see https://www.monetra.com/certifications.

action	Description	Telecheck	Telecheck IMG	Intuit/Echo ECC	Intuit/Echo IMG	Certegy FM1	Certegy FM2	Certegy ECC	Certegy ECC Img
checkverify	Verify Check	X		X		X	X		
checkconvonly	Convert Only			X					
checkconvvrfy	Convert with Verification	X		X					
checkconvguar	Convert with Guarantee			X				X	
checkconvover	Convert with Override			X					
checkimageupload	Upload check image		X		X				X
void	Remove transaction from batch	X		X	X			X	

6.4.8.2 Intuit/Echo

At settlement time, Monetra sends a Batch Retrieval with a request for summary data to Echo, which returns the balance Echo has on file. Monetra then stores this information in memory and issues a Batch Close. If this is successful and the balance returned from the summary request matches what Monetra has on file, the settlement returns AUTH and a phard_code of SUCCESS. If the balance does not match, then the phard_code will be BALANCE_MISMATCH and you must call the Echo help desk to determine where the mismatch lies.

6.4.8.3 Certegy

The merchant account setup for Certegy uses some different parameters:

- FM2MERCHID - Merchant ID for FM2 transactions
- DLCMERCHID - Merchant ID for check transactions with driver's license
- FM2MERCHIDBIZ - Same as FM2MERCHID, but for business checks
- DLCMERCHIDBIZ - Same as DLCMERCHID, but for business checks
- TERMTYPE - Special parameter that should only be used with specific instructions from Monetra support

6.5 Batch Management

6.5.1 Settle Batch

Submit the transactions in the batch for funding.

For most transaction types, this is a necessary second step to receive the authorized funds. To automate this step, see Section 5.10: Automated Merchant Task Management.

If you are using multiple subaccounts to settle different card types (e.g. sending Credit transactions to one processor and Gift transactions to another), you will need to specify a route in the sub parameter.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**settle**
batch	Y	N, Max 10	Batch number
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.5.2 Request Settlement Status

Request the settlement status of a batch in the processor's system.

This does not settle a batch. Typically, this queries the settlement status of the transactions in the batch in the processor's system and returns a settlement response.

If you are using multiple subaccounts to settle different card types (e.g. sending Credit transactions to one processor and Gift transactions to another), you will need to specify a route in the sub parameter.

Note: Not all processors support this action.

Access level: Section 2.4.2: Merchant User Authentication
Table legend: Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**settlerfr**
batch	Y	N, Max 10	Batch number
sub	C	N, Max 10	Subaccount to use (for split routes), or 0 to use the default route

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
See Appendix B: *Response Parameters* for more possible response parameters.		

6.6 Check Password

Verify the current user's password.

Access level:	Section 2.4.2: Merchant User Authentication
Table legend:	Section 3.4: Parameter Formatting Legend

Request Parameters			
Key	**Req**	**Spec**	**Description**
action	Y		**chkpwd**

Response Parameters		
Key	**Spec**	**Description**
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
pass_expire_secs	N, Max 19	Number of seconds until the password expires (or -1 for never or 0 for expired)

A Request Parameters

This appendix organizes and outlines the various parameters that can be sent with the Merchant Actions in Chapter 5: *Main Merchant User Actions* and Chapter 6: *Merchant Subuser Actions*.

A.1 Account Information

There are multiple ways to collect account information. These are the parameters specific to each entry method.

Note: Many of these parameters can be sent to Monetra encrypted when using an encrypting reader with a supported encryption method. In this case, the parameters are prepended with `e_`, e.g. `trackdata` becomes `e_trackdata`. An encrypted parameter is sent either in addition to or in place of the equivalent plaintext parameter.

Typically, integrators use UniTerm for device integration and do not work directly with encrypting readers. Due to the difference in encrypted reader output and some formats being proprietary, limited information about the encrypted account parameters is provided. If it is necessary to directly send encrypted reader data to Monetra, please contact support at support@monetra.com.

A.1.1 Chip Card Entry

Key	Req	Spec	Description
`icc`	Y	X	TLV data, as returned from an EMV device for a chip insert or tap
`rfid`	O	A	Indicates whether or not the payment was accepted via RFID (proximity/tap). Possible values: • `yes` • `no` • `capable` (POS is capable of accepting RFID, but this transaction was not performed in this manner)

A.1.2 Swiped Entry

Key	Req	Spec	Description
`trackdata`	Y	ANS	Magnetic stripe data, as read from card via swipe. May be Track 1 data, Track 2 data, or combined Track 1/Track 2, following this format: • Track 1 data must begin with `B` • If sending combined tracks, Track 1 data must begin with `%` and end with `?` • If sending combined tracks, Track 2 data must begin with `;` and end with `?` • No whitespace or `LRC` characters •

A.1.3 Keyed Entry

Key	Req	Spec	Description
account	Y	N	Account number printed on card
expdate	Y	N, 4	Expiration date printed on card, MMYY
cv	O	AN, Max 4	Card Verification value. Usually 3 digits on back of VISA/MC/Discover, or 4 digits on front of AMEX cards. Other possible values: • n - CV intentionally not provided (default if nothing sent). • nr - CV is present on card but not readable. • na - CV is not present on card.
cardholdername	O	ANS	Name as it appears on card/check
street	O	ANS	Street address
zip	O	AN	Zip code for AVS. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.

A.1.4 Token

Key	Req	Spec	Description
token	Y	N, Max 19	Token ID, as obtained from an Add Token request or from a Sale when tokenized=yes

A.1.5 CardShield Ticket

Key	Req	Spec	Description
cardshieldticket	Y	N, Max 9	Temporary CardShield ticket ID, as obtained from the CardShield subsystem

A.1.6 Bank Account (ACH)

Key	Req	Spec	Description
abaroute	Y	N, 9	ABA routing number from check
account	Y	N	Account number printed on card
accounttype	O	AS	Type of bank account. Possible values: • checking - Default, mutually exclusive with savings. • savings - Mutually exclusive with checking. • personal - Default, mutually exclusive with business. • business - Mutually exclusive with personal.
cardholdername	O	ANS	Name as it appears on card/check

A.1.7 Physical Check

While ACH data references a bank account and is valid as long as the account is, physical checks authorize a single transaction and are only good for that check number. Even though physical checks are converted into electronic checks during check conversion, the account data cannot be tokenized or reused via ttid.

Some processors require driver's license information in addition to account data. For these, you can either pass the trackdata from a swiped license (dltrack2) or the manually entered number and state (dlnumber and dlstate).

Note: To process payments using physical checks, you must use Check Conversion.

A.1.7.1 MICR Read

Key	Req	Spec	Description
micr	Y	ANS	TOAD data on a check, as read via a MICR reader
cardholdername	O	ANS	Name as it appears on card/check
dltrack2	O	ANS	AAMVA driver's license data, as read via magnetic swipe read
dlnumber	O	ANS	Number on driver's license
dlstate	O	N, 2	2-digit state code on driver's license
dob	O	N, 6	Date of birth, MMDDYY
ssn	O	N, 4	Last 4 digits of Social Security number
phone	O	N, Max 10	Customer's phone number, as NNNNNNNNNN

A.1.7.2 Manual Entry

Key	Req	Spec	Description
account	Y	N, 4-19	Account number from check
abaroute	Y	N, 9	ABA routing number from check
checknum	Y	N	Sequential number on check
cardholdername	O	ANS	Name as it appears on card/check
dltrack2	O	ANS	AAMVA driver's license data, as read via magnetic swipe read
dlnumber	O	ANS	Number on driver's license
dlstate	O	N, 2	2-digit state code on driver's license

A.2 Monetary Information

Key	Req	Spec	Description
amount	O	M	Total amount of transaction, including all subtotals and supplementary amounts. This is always the amount to be settled.
examount	O	M	Extra amount. Typically used for Retail and Restaurant tipping.
nsf	O	A	Boolean flag indicating whether or not the transaction should be approved with non-sufficient funds (NSF). For gift cards, defaults to yes. For all other cards, defaults to no. If this is disabled (set to no or not sent at all) and a partial approval is returned, the transaction is automatically reversed and an msoft_code of NSFAUTODENY is returned in the response. If this is enabled and there are not enough funds to cover the full requested amount, the transaction will be partially approved. In this case, the approved amount will be present in authamount in the response. When a partial authorization is received, a merchant should request another payment method for the remaining funds. It is acceptable to issue a reversal if no additional payment methods are possible.
cashbackamount	O	M	Amount of total given to customer as cash. Required if customer is receiving cash.
tax	O	M	Amount of total that is tax applied to the order, or nt if the transaction is non-taxable/tax-exempt
surchargeamount	O	M	Amount of total being received as a surcharge
currency	O	N	Currency code. If processing in a foreign currency, depending on the processor, this code might be required. ISO 4217 3-digit numeric code

A.3 Verification Information

Key	Req	Spec	Description
street	O	ANS	Street address
zip	O	AN	Zip code for AVS. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
cv	O	AN, Max 4	Card Verification value. Usually 3 digits on back of VISA/MC/Discover, or 4 digits on front of AMEX cards. Other possible values: • n - CV intentionally not provided (default if nothing sent). • nr - CV is present on card but not readable. • na - CV is not present on card.
fraudautodeny	O	A	Boolean flag indicating whether or not to override the user's default fraudautodeny settings. Defaults to no.

A.4 Order Information

Key	Req	Spec	Description
ordernum	O	AN, Max 25	Order number. Required for Ecomm/MOTO and Restaurant industries and for Level 2/3 qualification. It is recommended to always send this.
custref	O	ANS, Max 128	Customer reference number, as sent to the processing institution for Level II/III interchange qualification
ptrannum	O	N, Max 19	Similar to ordernum, but numeric for efficient indexing in the database. It is recommended to use ordernum instead.

A.5 PIN Information

Key	Req	Spec	Description
pin	O	ANX	Encrypted PIN block, as returned from PIN-entry device, or plaintext PIN for gift cards. For PINless debit bill payments, this should be set to pinless.
ksn	O	X, 16 or 20	Key serial number, as returned from PIN-entry device. If the left-most bytes are FFFF, those are padding and must be stripped.

A.6 Token Information

Key	Req	Spec	Description
tokenize	O		Boolean flag indicating whether or not the account data should be tokenized as part of this request. Defaults to no. The token will be of type store, which allows use with future financial transactions.
matching_token	O	A	With tokenize=yes, boolean flag indicating whether or not Monetra should first look for and use an existing token for this account before generating a new one. If multiple tokens for the same account are present, it is undefined which will be returned. Defaults to no.

A.7 COF/Recurring Information

Key	Req	Spec	Description
recurring	O	AS	Type of payment being processed with this stored card. Not necessary if using a stored token. Possible values: recurring - Payment is part of a series that will recur until end date is reached (or indefinitely, if no end date was specified). installment - Payment is part of a numbered series of payments, with the amount based on a fixed total divided by the number of installments. cardonfile - Payment is not based on a schedule. For customer-initiated transactions only. first - If payment is the first with this stored card, this should be pipe (|) separated with the type of series, e.g. recurring|first. A cof_transid and cof_authamount will then be returned, both of which should be passed in subsequent transactions if not using a stored token.
cof_transid	O	N	Transaction ID returned from the first COF transaction. Must be sent on all subsequent COF transactions. Not needed when referencing a stored token.
cof_authamount	O	M	Authorized amount returned from the first COF transaction. Must be sent on all subsequent COF transactions. Not needed when referencing a stored token.
installment_num	O	N, Max 2	For installment payments, the current payment number in a series of installments
installment_total	O	N, Max 2	For installment payments, the total number of payments to make

A.8 Processing Information

Key	Req	Spec	Description
cardpresent	O	A	Indicates whether or not the card is present with the merchant at the time of the transaction. Possible values: • yes - Card is present (default). • no - Card is not present, for phone/mail orders with a non-MOTO merchant account. • ecomm - Card is not present, for web orders with a non-ECOMM merchant account. • mobileinapp - Card is not present, for orders taken through an app, like Apple Pay.
capture	O	A	Boolean flag indicating whether or not the transaction will be added to the batch settlement. Defaults to yes.
debtrepayment	O	A	Boolean flag indicating whether or not the transaction is for repaying debt. There are regulations surrounding the use of this flag. Please consult your processor if it is appropriate to mark transaction as debt repayment based on your business type. Defaults to no.
duplcheck	O	A	Boolean flag indicating whether or not to enable duplicate checking. This overrides the default setting in the merchant configuration. The criteria for duplicate checking is stored within the merchant configuration. Defaults to no.
priority	O	A	Processing priority. Possible values: low normal high It is recommended to process large batches as priority low, so that any real-time transactions that come through are processed immediately.

Key	Req	Spec	Description
timeout	O	N	Time in seconds that transaction can sit in send queue before sending to the processor. This is a hint/estimate and not a hard value.
interchange_results	O	A	Boolean flag indicating whether or not to return additional interchange data received with the transaction. Defaults to no. These additional parameters are returned with the response (if the data is present in the transaction): • actcode • apcode • authsource • avsresp • raw_cardlevel • cavvresp • commind • cvresp • netident • posdata • proccode • raci • rrefnum • settledate • sqi • stan • token_requestor • token_assurance_level • trandate • tranflags • transid • trantime • valcode
txnexport	O	A	Indicates the transaction export status. See Section 5.11.1: Export Transaction for more information. Possible values: no - Do not export data (default). yes - Export unencrypted data. encrypted - Export encrypted data. Merchant User must have txn_export_key in config.
offline_decline	O	A	Reason the EMV device declined the transaction before going online. Possible values:

Key	Req	Spec	Description
			`chipmalfunction` - Device was unable to read chip on card. `badpin` - Card failed PIN validation. `carddecline` - Card declined the transaction.

A.9 Receipt Information

Key	Req	Spec	Description
rcpt	O	ANS	Receipt format. See Appendix C.9: Receipt Formats for specifications.
rcpt_email	O	ANS	Email address for sending receipt to customer, or a special value of customer to use the email address on file for the customer
rcpt_sms	O	N, 10	Phone number for texting receipt to customer, or a special value of customer to use the phone number on file for the customer. Merchant must have the allow_invoice_sms flag set.
rcpt_merch_copy	O	A	Boolean flag indicating whether or not to email a copy of the receipt to the merchant. This overrides the merchant flag receive_receipts_invoice.
rcpt_duplicate	O	A	Boolean flag indicating whether or not the receipt is a reprint. Defaults to no.

A.10 Shipping Information

Key	Req	Spec	Description
shipzip	O	AN	Zip code where product is being shipped. 5 or 9 digits for US zip codes, or 6 alphanumeric characters for Canadian postal codes.
shipcountry	O	AN	Country where product is being shipped, formatted as ISO 3166-1 numeric or alpha-3

A.11 Ecommerce Information

Key	Req	Spec	Description
cavv	O	ANS, Max 40	3-D Secure data. Base64-encoded CAVV/AAV response data from the VISA/MC authentication servers. If the card issuer does not use this system, the card issuer does but the cardholder does not, or the system is currently unavailable, send nonparticipant. Additionally, if the ECI is known (typically as part of the mobile in-app payload data), it can be prefixed to the CAVV data and pipe (|) separated.
3ds_txnid	O	ANS	EMV 3-D Secure (3DS 2.0) transaction ID generated by the Directory Server
goods	O	A	Type of product. Either physical (default) or digital.

A.12 Healthcare Information

Key	Req	Spec	Description
healthcare	O	A	Boolean flag indicating whether or not this is a healthcare transaction. Needed if the transaction is attempting to qualify for IIAS/ FSA/HRA. Defaults to no.
clinicamount	O	M	Amount of total that is for clinic-related services
dentalamount	O	M	Amount of total that is for dental-related services
otheramount	O	M	Amount of total that is for other qualified purchases (such as clinic expenses)
rxamount	O	M	Amount of total that is for prescriptions
transitamount	O	M	Amount of total that is for transportation
visionamount	O	M	Amount of total that is for vision-related services

A.13 Lodging Information

Key	Req	Spec	Description
bdate	O	ANS	Beginning date of stay. See Appendix C.8: Date Formats.
edate	O	ANS	Ending date of stay. See Appendix C.8: Date Formats.
excharges	O	AS	Additional charges not included in the room rate. See Appendix C.7: Extra Charge Codes.
rate	O	M	Room rate per night
roomnum	O	ANS	Room number
advancedeposit	O	A	Boolean flag indicating whether or not transaction is a deposit for a future stay. Defaults to no.
noshow	O	A	Boolean flag indicating whether or not charge is for a reservation where the customer did not show or canceled. Defaults to no.

A.14 Merchant Information

Key	Req	Spec	Description
`descmerch`	O	ANS	Merchant name descriptor, meant to change how a merchant is displayed on a cardholder's receipt. Formatting varies from processor to processor, and not all processors support this feature.
`descloc`	O	ANS	Merchant location descriptor. Formatting varies from processor to processor, and not all processors support this feature.

A.15 Lane Information

Key	Req	Spec	Description
laneid	O	N, Max 8	Unique lane/register identifier. It is recommended to always send this.
stationid	O	ANS	Free-form identifier for the physical station running the transaction
clerkid	O	ANS	Identifier for the clerk running the transaction, free-form
divisionnum	O	ANS	Description of division within company, usually for ACH transactions
comments	O	ANS	User-defined reporting field, free-form

A.16 Level III Information

Key	Req	Spec	Description
discountamount	O	M	Amount of discount applied to the order as a whole. If Level III line items are part of the order, this includes all discounts for each line item. The sum of all line item discounts must be less than or equal to this value.
dutyamount	O	M	Total amount of duty (fee associated with import of goods) on order
freightamount	O	M	Amount of freight/shipping on the order. For Level III, if shipping was charged on the order, this parameter must be sent. If the charge was greater than 0, then shipzip must also be sent. Freight is never part of the individual line items.

B Response Parameters

This appendix organizes and outlines the various parameters that can be received from the Merchant Actions in Chapter 5: *Main Merchant User Actions* and Chapter 6: *Merchant Subuser Actions*.

B.1 System Result Codes

These codes are returned with all transactions. The various system codes that are returned with transactions are used to determine the outcome of the transaction and provide additional details.

B.1.1 Authorization Codes

This is the absolute status of the transaction. All other codes are clarifying details.

This will appear in the response as `code`.

Code	Description
`AUTH`	Authorized/approved
`DENY`	Denied, and not likely to succeed on later attempts
`CALL`	Call processor for authorization
`CANCEL`	Canceled
`DUPL`	Duplicate transaction
`PKUP`	Confiscate card
`RETRY`	Temporary error, clerk should retry transaction
`SETUP`	Account setup error
`TIMEOUT`	No response received in set amount of time

B.1.2 Monetra Codes

This code is specific to Monetra and provides more information on any issue encountered within the system. This does not contain any information received from the processor.

This will appear in the response as `msoft_code`.

Result Code	Description
`INT_SUCCESS`	All local tests passed
`SNF`	Store and forward success
`INT_GENERICFAIL`	Generic or undefined failure
`UNKNOWN`	Unknown or unset, could be success or failure
`ACCT_ADMINTRANSNOTALLOWED`	Admin-level transactions not allowed on this port
`ACCT_AUTHFAILED`	Account authentication failed
`ACCT_DISABLED`	Account disabled
`ACCT_INVALIDTRANS`	Invalid transaction type for this user
`ACCT_OBSCUREREQUIRED`	Obscure restriction required for this request (see Section 5.1.2: Edit Merchant Subuser)

Result Code	Description
ACCT_PASSEXPIRED	Account password expired
ACCT_SSLCERT	SSL certificate check failed
ACCT_TOOMANYATTEMPTS	Too many bad login attempts
ACCT_TRANSNOTALLOWED	Missing necessary permissions for this request (see Section 5.1.2: Edit Merchant Subuser)
ACCT_USERTRANSNOTALLOWED	User-level transactions not allowed on this port
CONN_MAXATTEMPTS	Maximum attempts to connect to processor reached
CONN_MAXSENDS	Maximum attempts to send transaction reached
CONN_TOREVERSAL	Timeout Reversal issued, status of transaction unknown
DATA_ABAROUTE	Invalid ABA route (checks only)
DATA_ACCOUNT	Invalid account number
DATA_AMOUNT	Invalid amount
DATA_BADTRANS	Invalid transaction structure or data
DATA_BATCHLOCKED	Batch locked
DATA_EXPDATE	Invalid expiration date
DATA_INVALIDMOD	Invalid modification to existing transaction
DATA_MICR	Invalid MICR data (checks only)
DATA_NOOPENBATCHES	Batch not found
DATA_RECORDNOTFOUND	Record not found
DATA_TRACKDATA	Invalid track data
DB_FAIL	Failed to write to database
FRAUDAUTODENY	Transaction automatically denied due to fraud rule
NSFAUTODENY	Transaction automatically denied due to insufficient funds when the merchant did not allow partial approvals
LIC_CARDTYPE	Card type not allowed for this license
LIC_TRANEXCEED	Maximum number of transactions reached
LIC_USERS	Maximum number of user accounts reached
SETUP_CARDTYPE	Account not configured for this card type
SETUP_DATA	Generic configuration error
SETUP_SCHED	Transaction could not be scheduled
SETUP_TRANTYPE	Account not configured for this action

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

B.1.3 Processor Codes

This code is specific to the processing institution used for the transaction and is set according to data received from them.

This will appear in the response as phard_code.

Result Code	Description
SUCCESS	Generic success
UNKNOWN	Unknown reason
GENERICFAIL	Generic/undefined failure
ACCOUNT_CLOSED	Account closed
ACCTERROR	Account number or length error
ALREADY_ACTIVE	Gift card already activated
ALREADY_REVERSED	Reversal already issued
BAD_MERCH_ID	Invalid Merchant ID
BAD_PIN	Incorrect Debit/EBT PIN
BALANCE_MISMATCH	Settlement totals do not match
CALL	Call issuer for authorization
CARD_EXPIRED	Credit card expired
CASHBACK_EXCEEDED	Maximum cash back limit reached
CASHBACK_NOAVAIL	Cash back services not available
CID_ERROR	Invalid CVV2/CID
DATE_ERROR	Invalid date
DONOTHONOR	Do not honor card
DUPLICATE_BATCH	Duplicate batch number
ENCRYPTION_ERROR	Encryption error (usually for Debit/EBT)
EXCEED_ACTIVITY_LIMIT	Maximum activity limit reached
EXCEED_WITHDRAWAL_LIMIT	Maximum withdrawal limit reached
ID_ERROR	Valid ID required for transaction
INELIGIBLE_CONV	Check authorized but not eligible for conversion/ACH, keep check for deposit
INSUFFICIENT_FUNDS	Account does not have enough funds to cover the transaction

Result Code	Description
INVALID_ACCOUNT_TYPE	For Interac, invalid account type selection
INVALID_SERVICE_CODE	Invalid service code
MANAGER_NEEDED	Manager needed, possible velocity warning
NOREPLY	No response within processor's systems or beyond
NOT_ACTIVE	Account has not 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)
RECURRING_CANCEL	Recurring payment failed and will continue to fail
REENTER	Invalid transaction data or setup, reenter and retry
REJECTED_BATCH	Batch rejected for settlement
REPRESENTED	Duplicate check
RETRY	Retry the transaction
RETRY_FORCE_INSERT	Retry the transaction with ICC card inserted
SECURITY_VIOLATION	Security violation
SYSTEM_ERROR	Generic system error
VIOLATION	Generic system violation

B.2 Authentication Result Codes

The codes in this section are informational reports from the issuer to the merchant to help with verifying cardholders and reducing fraud.

Note: These codes alone do not approve or fail transactions; they are merely informational. It is up to the integrator to decide how to use these codes. To automatically deny transactions, see the `fraudautodeny` setting in `Add Merchant User` and `Edit Merchant User`.

B.2.1 AVS Result Codes

This will appear in the response as `avs`.

Result Code	Description
`GOOD`	All checks passed
`BAD`	All checks failed
`STREET`	Street verification failed
`ZIP`	Zip Code verification failed
`UNKNOWN`	Result unknown, should treat as good

B.2.2 CV Result Codes

This will appear in the response as `cv`.

Result Code	Description
`GOOD`	CV verification passed
`BAD`	CV verification failed
`UNKNOWN`	Result unknown, should treat as good

B.3 Card Level Result Codes

Visa card level results are returned from most processors and represent the product identifier/type of the Visa card used in the transaction.

This will appear in the response as `cardlevel` or `raw_cardlevel`.

LEGEND: ^ = space

Result Code	Raw Code	Description
VISA_TRADITIONAL	A^	Visa Traditional (US), Visa Classic (Canada)
VISA_TRADITIONAL_REWARDS	B^	Visa Traditional Rewards (US), Visa Gold or Visa Platinum (Canada)
VISA_SIGNATURE	C^	Visa Signature
VISA_SIGNATURE_PREFERRED	D^	Visa Signature Preferred (US), Visa Infinite (Canada)
PROPRIETARY_ATM	E^	Proprietary ATM
VISA_CLASSIC	F^	Visa Classic
VISA_BUSINESS	G^	Visa Business
VISA_CHECK	H^	Visa Consumer Check Card
VISA_INFINITE	I^	Visa Infinite
RESERVED	J^	Reserved
VISA_CORPORATE	K^	Visa Corporate
ELECTRON	L^	Visa Electron
MASTERCARD_EUROCARD_DINERS	M^	MasterCard/EuroCard and Diners
VISA_PLATINUM	N^	Visa Platinum
RESERVED	O^	Reserved
VISA_GOLD	P^	Visa Gold
PRIVATE_LABEL	Q^	Private Label
PROPRIETARY	R^	Proprietary
VISA_PURCHASE_CARD	S^	Visa Purchasing
RESERVED	T^	Reserved/Interlink
VISA_TRAVELMONEY	U^	Visa TravelMoney
VISA_VPAY	V^	V PAY
RESERVED	W^	Reserved
VISA_B2B_VIRTUAL_PAYMENTS	X^	Visa B2B Virtual Payments
RESERVED	Y^	Reserved
RESERVED	Z^	Reserved

Result Code	Raw Code	Description
RESERVED	0^	Reserved
RESERVED	1^	Reserved
RESERVED	2^	Reserved
RESERVED	3^	Reserved
RESERVED	4^	Reserved
RESERVED	5^	Reserved
RESERVED	6^	Reserved
RESERVED	7^	Reserved
RESERVED	8^	Reserved
RESERVED	9^	Reserved
VISA_SIGNATURE_BUSINESS	G1	Visa Signature Business
VISA_BUSINESS_CHECK	G2	Visa Business Check
VISA_BUSINESS_ENHANCED	G3	Visa Business Enhanced/Visa Platinum Business
VISA_BUSINESS_INFINITE	G4	Visa Infinite Business
VISA_BUSINESS_REWARDS	G5	Visa Business Rewards
VISA_INFINITE_PRIVILEGE	I1	Visa Infinite Privilege
VISA_ULTRA_HIGH_NET_WORTH	I2	Visa Ultra High Net Worth
VISA_GENERAL_PREPAID	J1	Visa General Prepaid
VISA_PREPAID_GIFT	J2	Visa Prepaid Gift Card
VISA_PREPAID_HEALTH	J3	Visa Prepaid Healthcare
VISA_PREPAID_COMMERCIAL	J4	Visa Prepaid Commercial
VISA_GSA_CORPORATE_TANDE	K1	Visa GSA Corporate T&E
VISA_REWARDS	N1	Visa Rewards
VISA_SELECT	N2	Visa Select
PRIVATE_LABEL_PREPAID	Q1	Private Label Prepaid
VISA_PRIVATE_LABEL_BASIC	Q2	Private Label Basic
VISA_PRIVATE_LABEL_STANDARD	Q3	Private Label Standard
VISA_PRIVATE_LABEL_ENHANCED	Q4	Private Label Enhanced
VISA_PRIVATE_LABEL_SPECIALIZED	Q5	Private Label Specialized
VISA_PRIVATE_LABEL_PREMIUM	Q6	Private Label Premium
VISA_PURCHASE_FLEET	S1	Visa Purchasing with Fleet
VISA_GSA_PURCHASE	S2	Visa GSA Purchasing
VISA_GSA_PURCHASE_FLEET	S3	Visa GSA Purchasing with Fleet
VISA_COMMERCIAL_AGRICULTURE	S4	Visa Commercial Agriculture

Result Code	Raw Code	Description
VISA_COMMERCIAL_TRANSPORT	S5	Visa Commercial Transport
VISA_COMMERCIAL_MARKETPLACE	S6	Visa Commercial Marketplace
RESERVED	V1	Reserved
AMEX	AX	American Express
DISCOVER	DI	Discover
DINERS_CLUB	DN	Diners Club International
JCB	JC	JCB

B.4 Alphabetical Listing

This is an alphabetical listing of the various parameters that can be received from the Merchant Subuser Actions in Chapter 6: *Merchant Subuser Actions*. Not all parameters are sent back for all transactions. The actual parameters returned from a transaction depend on various factors, such as the transaction type, processor used, industry, parameters sent, parameters requested, etc. This list is meant to be an easy reference for better understanding returned data.

Key	Spec	Description
abaroute	N, 9	ABA routing number from check
account	N	Masked account number
accounttype	AS	Type of bank account. See Appendix A.1.6: Bank Account (ACH).
actcode	ANS	Action Code. See Appendix B.5.1: Action Code.
apcode	AN	Approval Code. See Appendix B.5.2: Approval Code.
auth	AN	Approval Code as returned from processor. See Appendix B.5.2: Approval Code.
authamount	M	Amount of funds actually authorized. This is only present when the authorized amount is less than the requested amount, as can happen with insufficient funds. If all of the requested amount is authorized, then this will not appear in the response.
authsource	AN, Fixed 1	Authorization Source Code. See Appendix B.5.3: Authorization Source Code.
avs	A	Address verification result. See Appendix B.2.1: AVS Result Codes.
balance	M	Balance remaining on card
batch	N, Max 10	Batch number
batconnum	ANS	Some processors return an element similar to an authorization number when you settle a batch
cardholdername	ANS	Name as it appears on card/check
cardlevel	AN, Max 2	Card level result. See Appendix B.3: Card Level Result Codes.
cardtype	A	Card type detected. See Appendix C.1: Card Types.

Key	Spec	Description
cavvresp	AN, Fixed 1	Raw code from processor indicating CAVV result. See Appendix B.5.6: CAVV Result Code.
code	A	Transaction result code. See Appendix B.1.1: Authorization Codes.
cof_authamount	M	Authorized amount returned from the first COF transaction. Must be sent on all subsequent COF transactions. Not needed when referencing a stored token.
cof_transid	N	Transaction ID returned from the first COF transaction. Must be sent on all subsequent COF transactions. Not needed when referencing a stored token.
commind	AN, Fixed 1	Commercial Card Response Indicator. See Appendix B.5.7: Commercial Card Response Indicator.
custref	ANS, Max 128	Customer Reference Number
custvatnum	ANS	Customer VAT registration number
cv	A	Cardholder verification result. See Appendix B.2.2: CV Result Codes.
cvresp	A, Fixed 1	Raw code from processor indicating CV result. See Appendix B.5.8: Cardholder Verification Response Code.
entrymode	A, Fixed 1	Monetra-defined value indicating how the transaction data was entered. Possible values: • C - Chip read • F - EMV Fallback to swipe • G - EMV Fallback to keyed • I - MICR read • M - Manually entered • R - MSD Tap • S - Swipe • T - EMV Tap
esi	ANS, Fixed 3	Ecommerce Security Indicator, for Mastercard 3-D Secure transactions
icc	X	TLV data for EMV device
item	N	Sequence number
laneid	N, Max 8	Unique lane/register identifier

Key	Spec	Description
language	A	2-character language code. Possible values: en - English fr - French es - Spanish de - German it - Italian
msoft_code	ANS	Detailed result code specific to system-internal checks. See Appendix B.1.2: Monetra Codes.
netident	AN, Fixed 1	Network Identification Code. See Appendix B.5.10: Network Identification Code.
orig_authamount	M	authamount from the original transaction
pclevel	N	Card level. Possible values: 0 - Consumer card 1 - Commercial card 2 - Corporate or government purchase card
phard_code	AS	Detailed result code for success/fail from processor. See Appendix B.1.3: Processor Codes.
posdata	ANS	POS Data. See Appendix B.5.11: POS Data Code.
proccode	AN, Fixed 6	Processing Code. See Appendix B.5.12: Processing Code.
raci	AN, Fixed 1	Return Authorization Characteristics Indicator. See Appendix B.5.13: Returned Authorization Characteristics Indicator.
raw_avs	AN, Fixed 1	Raw code from processor indicating AVS result. See Appendix B.5.4: AVS Response Code.
raw_cardlevel	ANS, Fixed 2	Raw code from processor indicating card level according to Visa 62.23 standard. See Appendix B.3: Card Level Result Codes.
raw_code	AN	Raw code from processor indicating result of transaction. Each processor has different specifications for this value.
raw_cv	A, Fixed 1	Raw code from processor indicating CV result. See Appendix B.5.8: Cardholder Verification Response Code.
rcpt_*	ANS	Multiple possible response parameters. See Appendix C.9: Receipt Formats.

Key	Spec	Description
rrefnum	AN, Max 12	Retrieval Reference Number. See Appendix B.5.14: Retrieval Reference Number.
sequenceid	N	For denied transactions, sequence number within Monetra
settledate	N, Fixed 4	Settlement Date as MMDD, as returned by the processor. Debit/EBT cards only.
sqi	AN	Spend Qualified Indicator (SQI) or Transaction Integrity Class (TIC). See Appendix B.5.15: Spend Qualified Indicator.
stan	AN	System Trace Audit Number. See Appendix B.5.16: System Trace Audit Number.
timestamp	N, Max 19	Unix timestamp of transaction (seconds since January 1, 1970)
token_assurance_level	AN, Max 2	For Mastercard, Token Assurance Level
token_requestor	ANS, Max 11	For Mastercard, Token Requestor
trandate	N, Fixed 6	Processor-set Transaction Date, MMDDYY
tranflags	ANS	Monetra-defined Transaction Flags. See Appendix B.5.18: Transaction Flags.
transid	ANS, Fixed 15	Transaction ID. See Appendix B.5.17: Transaction ID.
trantime	N, Fixed 6	Processor-set Transaction Time, HHMMSS
ts	N, Max 19	For Debit/EBT, Authorization Timestamp, in seconds since the Unix epoch
ttid	N, Max 19	Unique ID referencing a transaction within Monetra
valcode	AN, Fixed 1	Validation Code. See Appendix B.5.19: Validation Code.
verbiage	ANS	Human-interpretable text of transaction outcome, meant to be passed on to display
wallet_identifier	ANS, Max 3	For Mastercard, Wallet Identifier

B.5 Raw Response Codes

These codes come straight back from the processor without any interpretation from Monetra.

To see these additional parameters in the response, you would either send interchange_results=yes with a normal transaction request or use a **trandetail** request after an authorization.

B.5.1 Action Code

Indicates the outcome of the transaction and the action that should be taken. Each processor sets their own specifications for the meaning of these codes.

This will appear in the response as actcode.

B.5.2 Approval Code

This is the code for referencing a specific authorization. It is typically 6 digits, but some processors will return other values during testing.

This will appear in the response as apcode.

Visa, Mastercard, and Discover use the last position of the approval code for card product identification:

Raw Code	Description
Visa	
A	Visa Traditional (US), Visa Classic (Canada)
B	Visa Traditional Rewards (US) Visa Gold or Visa Platinum (Canada)
C	Visa Signature
D	Visa Signature Preferred (US), Visa Infinite (Canada)
Mastercard	
Z	Default value indicating that the specific cardholder account does not participate in Account Level Management processing
B	Enhanced Value Platform (Consumer)
C	Product Graduation Consumer World
D	Product Graduation Consumer World Elite
G	Product Graduation Business World
H	Product Graduation Business World Elite
I	Product Graduation Corporate World
J	Product Graduation Corporate World Elite
M	Enhanced Value Platform and Product Graduation
P	Product Graduation Only

Raw Code	Description
S	High Value
T	Product Graduation and High Value
Discover	
C	Consumer Credit Core Card
R	Consumer Credit Rewards Card
P	Consumer Credit Premium Card
Q	Consumer Credit Premium Plus Card
B	Commercial Credit Card
E	Commercial Executive Business Credit Card
Z	Unspecified Card Product

B.5.3 Authorization Source Code

Indicates the source of the authorization.

This will appear in the response as authsouce.

Raw Code	Description
1	Stand-in time-out response
2	Stand-in amount below issuer limit
3	Stand-in issuer in Suppress Inquiry mode
4	Direct connect issuer generated response (MasterCard, American Express), or Stand-in: issuer unavailable
5	Issuer Generated Response
6	Offline approval, POS device generated
7	Acquirer approval: Base I unavailable
8	Acquirer approval of a referral
9	Use for non-authorized transactions; such as credit card credits
D	Referral: authorization code manually keyed
E	Offline approval: authorization code manually keyed
F	CAFIS Interface Offline Post-Auth (currently in use by Japan Acquirer Services [JAS])
G	Issuer Approval, Post-Auth

B.5.4 AVS Response Code

Indicates the exact result of the AVS check.

This will appear in the response as avsresp or raw_avs.

Raw Code	Description
0	Address verification was not requested
A	Address match only
B	Street Address match for international transaction. Postal Code not verified because of incompatible formats (Acquirer sent both street address and Postal Code)
C	Street address and Postal Code not verified for international transaction because of incompatible formats (Acquirer sent both street address and Postal Code)
D	Street address and postal code match for International transaction
F	UK only. Street address and Postal Code match
G	Non-US issuer does not participate
I	Address information not verified for international transaction
M	Street address and postal code match for international transaction
N	No address or street match
P	Postal Codes match for international transaction. Street address not verified because of incompatible formats
R	Issuer system unavailable - Retry
S	Service not supported
U	Address unavailable
W	Nine character numeric Zip match only (street/address failed)
X	exact match, nine character numeric Zip
Y	exact match, five character numeric zip
Z	five character numeric zip match only (street/address failed)
1	Amex only. Cardholder Name and ZIP match
2	Amex only. Cardholder Name, address, and ZIP match
3	Amex only, Cardholder Name and address match
4	Amex only. Cardholder Name match
5	Amex only. Cardholder name incorrect, ZIP match
6	Amex only. Cardholder Name incorrect, address and ZIP match
7	Amex only. Cardholder Name incorrect, address match
8	Amex only. All do not match

B.5.5 Card Level Code

Indicates the branding of the card. This is only relevant for Visa cards and cards routed through the Visa network. The standards are defined by Visa 62.23.

This will appear in the response as `cardlevel`.

LEGEND: ^ = space

Raw Code	Description
`A^`	Visa Traditional (US), Visa Classic (Canada)
`B^`	Visa Traditional Rewards (US), Visa Gold or Visa Platinum (Canada)
`C^`	Visa Signature
`D^`	Visa Signature Preferred (US), Visa Infinite (Canada)
`E^`	Proprietary ATM
`F^`	Visa Classic
`G^`	Visa Business
`H^`	Visa Consumer Check Card
`I^`	Visa Infinite
`J^`	RESERVED
`K^`	Visa Corporate
`L^`	Visa Electron
`M^`	MasterCard/EuroCard and Diners
`N^`	Visa Platinum
`O^`	RESERVED
`P^`	Visa Gold
`Q^`	Private Label
`R^`	Proprietary
`S^`	Visa Purchasing
`T^`	Reserved/Interlink
`U^`	Visa TravelMoney
`V^`	V PAY
`W^`	RESERVED
`X^`	Visa B2B Virtual Payments
`Y^`	RESERVED
`Z^`	RESERVED
`0^`	RESERVED
`1^`	RESERVED
`2^`	RESERVED
`3^`	RESERVED
`4^`	RESERVED
`5^`	RESERVED
`6^`	RESERVED
`7^`	RESERVED

Raw Code	Description
8^	RESERVED
9^	RESERVED
G1	Visa Signature Business
G2	Visa Business Check
G3	Visa Business Enhanced/Visa Platinum Business
G4	Visa Infinite Business
G5	Visa Business Rewards
I1	Visa Infinite Privilege
I2	Visa Ultra High Net Worth
J1	Visa General Prepaid
J2	Visa Prepaid Gift Card
J3	Visa Prepaid Healthcare
J4	Visa Prepaid Commercial
K1	Visa GSA Corporate T&E
N1	Visa Rewards
N2	Visa Select
Q1	Private Label Prepaid
Q2	Private Label Basic
Q3	Private Label Standard
Q4	Private Label Enhanced
Q5	Private Label Specialized
Q6	Private Label Premium
S1	Visa Purchasing with Fleet
S2	Visa GSA Purchasing
S3	Visa GSA Purchasing with Fleet
S4	Visa Commercial Agriculture
S5	Visa Commercial Transport
S6	Visa Commercial Marketplace
V1	RESERVED
AX	American Express
DI	Discover
DN	Diner's Club International
JC	JCB

B.5.6 CAVV Result Code

Indicates the result of the validation check of the Cardholder Authentication Verification Value (CAVV).

CAVV is the method Visa uses to generate the data for 3-D Secure transactions and is the standard name Monetra uses for this parameter. It is not limited to any card brand.

This will appear in the response as `cavvresp`.

Raw Code	Description
N/A	not validated
`0`	Not validated due to erroneous data
`1`	Failed validation
`2`	Passed validation
`3`	Could not be performed. Issuer attempt incomplete
`4`	Could not be performed. Issuer system error
`5`	Reserved
`6`	Reserved
`7`	Failed validation. US card, non-US acquirer
`8`	Passed validation. US card, non-US acquirer
`9`	Failed validation, issuer unavailable. US card, non-US acquirer
`A`	Passed validation, issuer unavailable. US card, non-US acquirer
`B`	Passed validation. Informational only, no liability shift
`C`	Not validated, attempted
`D`	Not validated, authentication

B.5.7 Commercial Card Response Indicator

Indicates the commercial/purchase level of the card. Monetra uses this to determine the card's `pclevel`.

This will appear in the response as `commind`.

Raw Code	Description	`pclevel`
`B`	Business Card	1
`D`	Visa Commerce (reserved)	1
`R`	Corporate Card	1
`S`	Purchasing Card	2
`0`	Non-commercial Card	0
space	Invalid indicator received	0

B.5.8 Cardholder Verification Response Code

Indicates the outcome of the CV check.

This will appear in the response as `cvresp` or `raw_cv`.

Raw Code	Description
M	Match
N	No Match
P	Not Processed
S	Merchant has indicated that Verification Code is not present on card
U	Issuer is not certified and/or has not provided Visa encryption keys

B.5.9 Issuer Response Code

Response code from the card issuer (not the processor).

This will appear in the response as `issuerrespcode`.

B.5.10 Network Identification Code

Debit/EBT only. Indicates the network on which the transaction was authorized.

This will appear in the response as `netident`.

Raw Code	Description
V	VisaNet
G	Interlink
B	Plus ATM
O	Cirrus ATM
J	Mastercard ATM
N	STAR (formerly Cash Station)
S	PULSE (formerly Money Station)
W	STAR SE
Z	STAR NE
Q	STAR W
U	AFFN
M	STAR
8	Maestro
L	Pulse

Raw Code	Description
Y	NYCE
H	PULSE (formerly TYME)
E	Accel
P	NETS
C	CU24
3	Alaska Option
F	NYCE
7	ITS Shazam
K	EBT
T	EBT ATM
A	Amex ATM
D	Discover ATM
1	AFFN ATM

B.5.11 POS Data Code

Indicates the state of the POS at the time of the transaction.

Monetra generates this for the transaction, and it is sometimes sent back by the processor. The various values in the data are unique to each processor.

This will appear in the response as `posdata`.

B.5.12 Processing Code

Discover only, indicates the type of authorization request submitted.

This will appear in the response as `proccode`.

B.5.13 Returned Authorization Characteristics Indicator

For Visa and Mastercard, indicates additional information about the POS.

This will appear in the response as `raci`.

Raw Code	Description
A	Card Present qualified
C	Card Present qualified for a self-service automated fuel dispense
E	Card Present qualified and Card Acceptor Data was supplied in the authorization request

Raw Code	Description
F	Card Present qualified for visa account funding transactions
I	Card Present qualified incremental authorization request
K	Card Present qualified and included an address verification request in the authorization request (Unable to read magnetic stripe)
M	Meets national payment service requirements with no address verification: Direct Marketing
N	Not qualified
P	Card Present qualified and accepted for Preferred Customer qualification or 3D Secure validation failure for T&E transaction
R	Recurring or Installment payments
S	Card Present attempted for Preferred Ecommerce (3D Secure)
T	Transaction cannot participate in Card Present
U	Card Present qualified for Preferred Ecommerce (3D Secure)
V	Card Present qualified and included an address verification request in the authorization request
W	Card Present qualified for Basic Ecommerce (Non-3D Secure)
space	If Y sent and transaction not qualified (VAS downgrade)

B.5.14 Retrieval Reference Number

Debit/EBT only. Returned by the authorizing system.

This will appear in the response as `rrefnum`.

B.5.15 Spend Qualified Indicator

Returned for Visa and Mastercard transactions. Mastercard refers to this as the Transaction Integrity Class.

This will appear in the response as `sqi`.

Raw Code	Description
Visa	
B	Base spend assessment threshold has been met.
N	Spend qualification threshold has not been met.
Q	Spend qualification threshold has been met.
Mastercard	
A1	EMV/Token in a Secure, Trusted environment (Card Present)
B1	EMV/Chip Equivalent (Card Present)
C1	MagStripe (Card Present)
E1	Key Entered (Card Present)
U0	Unclassified (Card Present)
A2	Digital Transaction (Card and/or CardHolder Not Present)
B2	Authenticated Checkout (Card and/or CardHolder Not Present)
C2	Transaction Validation (Card and/or CardHolder Not Present)
D2	Enhanced Data (Card and/or CardHolder Not Present)
E2	Generic Messaging (Card and/or CardHolder Not Present)
U0	Unclassified (Card and/or CardHolder Not Present)

B.5.16 System Trace Audit Number

Debit/EBT and Discover only. A unique host-assigned identification number.

This is typically used for printing on receipts. If necessary, this should be preferred over the `rrefnum`.

This will appear in the response as `stan`.

B.5.17 Transaction ID

Unique identifier set by the card brand.

This will appear in the response as `transid`.

The different brands specify different meanings for this value, but the value is always 15 characters long.

Card Brand	Description
Visa	Transaction Identifier
Mastercard	4-character BankNet Date + 9-character BankNet Reference Number + 2 spaces
American Express	Transaction Identifier
Discover	Network Reference ID (NRID)
Debit/EBT/ Gift Card	Issuer-assigned identifier

B.5.18 Transaction Flags

Flags indicating various details about the transaction.

This will appear in the response as `tranflags`.

Flag	Description
ACCT_BUSINESS	For Check transactions, the account is a business account (not present indicates personal account)
ACCT_SAVINGS	For Check transactions, the account is a savings account (not present indicates checking account)
ADVANCEDEPOSIT	For Lodging, the transaction is being run for a future reservation
CARDNOTPRESENT	Card was not present at the merchant's location when the transaction was run
CARDONFILE	Account data came from a stored token (customer-initiated transactions only)
CARDSHIELD_E2E	Monetra CardShield end-to-end encryption was used
CAVVBAD	3D Secure authentication was attempted but unsuccessful
CAVVFULL	3D Secure authentication was performed successfully
CNP_ECOMM	For non-Ecomm industries, the transaction was run in an Ecomm setting
DATAKEYED	Account data was entered manually
DATAMICR	Account data was read by a MICR reader
DATATRACK1	Track 1 data is present
DATATRACK2	Track 2 data is present
DEBTREPAYMENT	The transaction is for paying off debt
DIGITAL_GOODS	The merchandise is a digital product
EMV_FALLBACK	An EMV transaction was attempted, but a different entry method had to be used

Flag	Description
EMV_FALLBACK_NOAID	An EMV transaction was attempted, but a different entry method had to be used because the merchant does not accept that type of card
EMV_OFFLINE	The transaction was approved by the card without going online to the issuer
ENCPROV_E2E	End-to-end encryption with an external provider was used
HASAVS	AVS data is present
HEALTHCARE	The transaction is for a healthcare-related payment
ICC	The transaction has EMV data from a chip card
INCREMENTAL	One or more incremental transactions was performed
INSTALLMENT	The transaction is one in a number of installment payments
MOBILEINAPP	The transaction took place in an app on a mobile device
NONTAXABLE	The transaction is tax-exempt
NOSHOW	For Lodging, the cardholder did not show up for the reservation
NSF	The transaction can be approved for less than the requested amount (non-sufficient funds)
PARTIALAUTH	The transaction was approved for less than the amount requested (non-sufficient funds)
PINBYPASS	For EMV transactions, a PIN was deliberately not entered
PINLESS	A Debit card was used, but a PIN was not entered
PROC_E2E	Processor-level end-to-end encryption was used
PROC_TOKEN	The account number on file is really a processor token
RECURRING	The transaction is a recurring payment
RECURRING_FIRST	The transaction is the first in a series of recurring or installment payments
RFID	Account data was read using a contactless RFID device/interface
RFIDCAPABLE	The device is capable of contactless transactions
SNF_APPROVED	The transaction was stored for later authorization, and the authorization was approved
SNF_DENY	The transaction was stored for later authorization, but the authorization was denied
STANDIN	Transaction was stored for later authorization

B.5.19 Validation Code

Visa, Mastercard, and Discover only. Indicates various details about the transaction's data.

Visa's value contains specific information generated by the card issuer and should not be interpreted by the POS device. For Mastercard and Discover's values, see the tables below.

This will appear in the response as `valcode`.

B.5.19.1 Mastercard

Position 1: CVC3 Error Indicator. Possible values:

Raw Code	Description
`Y`	CVC3 incorrect
`N`	CVC3 okay
`P`	CVC3 could not be validated
`E`	Length of unpredictable number was not valid

Position 2: POS Entry Mode Change. Possible values:

Raw Code	Description
`Y`	Mastercard changed POS-only mode from `02` to `09`, and Member is in monitoring mode
`N`	POS entry mode was not changed

Position 3: Transaction Edit Error Code/Magnetic Strip Quality Indicator. This indicates an error existed in the original authorization data.

Position 4: Filler

B.5.19.2 Discover

Position 1: Track 1 data indicator. Possible values:

Raw Code	Description
`0`	No Track 1 present
`1`	Track 1 data present with CVV not provided
`2`	Track 1 data present with non-zero and non-blank CVV
`3`	Track 1 data present with CVV set to all zeros
`4`	Track 1 data present with CVV containing some or all blanks
`5`	Track 1 data present but CVV location not disclosed by issuer

Position 2: Track 2 data indicator. Possible values:

Raw Code	Description
`0`	No Track 2 data present
`1`	Track 2 data present with CVV not provided
`2`	Track 2 data present with non-zero and non-blank CVV
`3`	Track 2 data present with CVV set to all zeros
`4`	Track 2 data present with CVV containing some or all blanks
`5`	Track 2 data present but CVV location not disclosed by issuer

Positions 3 and 4: POS Entry Mode

C General Codes and Formats

This appendix contains tables for various codes that you can encounter while communicating with Monetra.

C.1 Card Types

These are the card types that Monetra supports.

Type	Category	Description
VISA	Credit	Visa
MC	Credit	Mastercard
AMEX	Credit	American Express
DISC	Credit	Discover
DINERS	Credit	Diners Club
CUP	Credit	China UnionPay
JCB	Credit	Japan Commerce Bank
CB	Credit	Carte Blanche
SWITCH	Credit	Switch/Solo
BML	Credit	Bill Me Later
PAYPALEC	Credit	PayPal Express Checkout
VISADS	Debit -> Credit	Visa debit card that the processor actually ran as a credit transaction
VISADEBIT	Debit	Debit card backed by Visa
MCDEBIT	Debit	Debit card backed by Mastercard
OTHERDEBIT	Debit	Debit card not backed by brand
INTERAC	Debit	Interac (Canadian debit network)
EBT	EBT	EBT (Electronic Benefits Transfer -- Food Stamps/Cash Benefits)
CHECK	Check	Electronic Check
ACH	Check	US ACH Network
GIFT	Gift	Gift Card
OTHER	Gift	Generic Gift/Loyalty
ALL	Credit,Debit,EBT,Check,Gift	Aggregate macro for all card types
ALLCREDIT	Credit	Aggregate macro for all credit card types
USDEBIT	Debit	Aggregate macro for all US debit 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 (currently only one type though)
UNKNOWN	N/A	Unknown card type

C.2 EMV Entry Modes

These are the chip card (ICC) interfaces that Monetra supports.

Type	Category	Description
VISA_CONTACT	Credit	Visa, Contact
VISA_CONTACTLESS	Credit	Visa, Contactless
MC_CONTACT	Credit	Mastercard, Contact
MC_CONTACTLESS	Credit	Mastercard, Contactless
AMEX_CONTACT	Credit	American Express, Contact
AMEX_CONTACTLESS	Credit	American Express, Contactless
DISC_CONTACT	Credit	Discover, Contact
DISC_CONTACTLESS	Credit	Discover, Contactless
CUP_CONTACT	Credit	China UnionPay, Contact
CUP_CONTACTLESS	Credit	China UnionPay, Contactless
JCB_CONTACT	Credit	Japan Commerce Bank, Contact
JCB_CONTACTLESS	Credit	Japan Commerce Bank, Contactless
SWITCH_CONTACT	Credit	Switch/Solo, Contact
SWITCH_CONTACTLESS	Credit	Switch/Solo, Contactless
USDEBIT_CONTACT	Debit	US Debit, Contact
USDEBIT_CONTACTLESS	Debit	US Debit, Contactless
INTERAC_CONTACT	Debit	Interac (Canadian Debit), Contact
INTERAC_CONTACTLESS	Debit	Interac (Canadian Debit), Contactless
INTLDEBIT_CONTACT	Debit	International Debit, Contact
INTLDEBIT_CONTACTLESS	Debit	International Debit, Contactless

C.3 EMV Terminal Capabilities

These are the chip card (ICC) terminal capabilities that Monetra supports. When configuring a merchant account, these must match the capabilities of the device in the certification list in the UniTerm guide.

Capability	Description
ONLINEPIN	Allow the card to verify the PIN by going online
NOOFFLINEPIN	Do not allow the card to verify the PIN without going online
NOSIG	Do not allow signature
CASHBACK	Allow cash back or cash withdrawal
PINBYPASSCREDIT	Allow bypassing the PIN prompt for Credit Cards (card configuration must support another CVM)
PINBYPASSUSDEBIT	Allow bypassing the PIN prompt for Debit Cards (card configuration must support another CVM)

C.4 Processor Features

These are the various features that a processor module can support.

Feature	Description
NONE	No features
BIGBATCH	Capable of multi-merchant settlement. See Section 4.11: Big Batch Aggregation.
DEPRECATED	Deprecated processor module
EMULATION	Emulated processor
EMV	Supports EMV/Chip transactions
E2E	Capable of processor-level end-to-end encryption
HEALTHCARE	Supports healthcare transactions
HOSTCAPTURE	Has control over the batch
LEVEL2	Supports Level 2 processing
LEVEL3	Supports Level 3 line item processing
MULTICURRENCY	Supports international currencies
PINLESSDEBIT	Supports PINless Debit transactions
PREPAIDGIFT	Supports prepaid branded gift cards
SPLITAUTH	Supports settling with a different processor
SPLITSETTLE	Allows authorizations from a different processor
STANDIN	Supports stand-in (store and forward) authorizations
TERMLOADING	Capable of online terminal loading
TOKENIZATION	Supports tokenization
VBV_AND_MSC	Supports 3-D Secure

C.5 Industry Codes

These codes are used to specify the industry in which the merchant processes transactions.

Code	Description
AF	Automated Fueling
E	Ecommerce (ECOM)
F	Restaurant
G	Grocery
H	Hotel/Lodging
M	Mail-Order/Telephone-Order (MOTO)
R	Retail
RS	Retail Self-Service (Kiosk)

C.6 Connectivity Modes

These are the various connectivity modes that a processor module can support.

Code	Description
SSL	
HTTPS	
IP	
DIAL	
OTHER	

C.7 Extra Charge Codes

These are the codes used to indicate what extra charges were applied to the room for Lodging transactions. Multiple values should be pipe-separated (|).

Note: These are used to indicate what extra charges were applied, not how much or for what items.

Result Code	Description
REST	Restaurant/Room Service charges
GIFT	Gift Shop charges
MINI	Mini Bar charges
TELE	Telephone charges
LAUND	Laundry charges
OTHER	Other charges

C.8 Date Formats

These are the date formats for various parameters, such as `bdate` and `edate`. You can use either an absolute date, a relative date, or a special keyword.

C.8.1 Absolute Dates

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

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

C.8.2 Relative Dates

A relative date takes this format:
[+|-][amount] [unit]

Units
year[s]
week[s]
day[s]
hour[s]
min[s] / minute[s]
sec[s] / second[s]

Example:

```
+1 day
-5 years
```

C.8.3 Special Keywords

Name	Meaning
now	Current date/time
epoch	Unix timestamp (beginning on Jan 1, 1970 00:00:00 UTC)

C.9 Receipt Formats

Note: For much greater details on receipts--including how the receipt blocks appear in responses and details on building an actual receipt--see the relevant appendix on receipt formatting in the UniTerm Guide [https://www.monetra.com/developers].

The optional rcpt parameter is sent with transactions to indicate receipt formatting requirements. Each part is a key/value pair indicating formatting configurations, with the parts separated by semicolons, as in these examples:

```
rcpt=type=plain;line_len=24;use_merch_lang=no;line_break="\n"
rcpt=type=plain|html
```

These are the available keys:

Key	Description
type	List of receipt output formats to use, separated by pipes (|). When specifying more than one type, the response keys will indicate the format in the key name. If only one type is specified, the type is omitted. Possible values: plain - plain text (default) html - HTML, needs style sheet applied xml - XML, suitable for XSLT transformations, typically used to generate complex HTML when CSS alone is not capable of providing the desired formatting json - JSON, typically used for easy manipulation with JavaScript
line_len	Only relevant for type=plain. Number of characters per line. Default is 40.
line_break	Only relevant for type=plain. Character sequence for newlines. Default is \r\n.
use_merch_lang	Boolean flag indicating whether or not to use the merchant's selected language rather than the cardholder's language for the receipt. Defaults to yes.

C.10 BIN Range Format

BIN ranges are specified using a formula that details exactly what values are allowed.

An individual range is specified using this formula:
`min_prefix[-max_prefix][:min_len[-max_len][:cardtype[:pclevel]]]`

Multiple ranges are joined together with semicolons (`;`).

Format	Req	Spec	Description
min_prefix	Y	N	Minimum prefix that the account number must start with
`max_prefix`	O	NS	Maximum prefix that the account number can start with. Must be the same number of digits as `min_prefix`.
`min_len`	O	N	Minimum number of digits the account number must have
`max_len`	O	NS	Maximum number of digits the account number can have
`cardtype`	O	A	Card type to match. See Appendix C.1: Card Types.
`pclevel`	O	N	Card level to match. See Appendix B.4: Alphabetical Listing. Note that the range is from 0 to 2.

Example ranges:

Range	Meaning
`30`	Match all accounts that start with `30`
`30-33`	Match all accounts that start with a value between `30` and `33`
`30-33:15`	Match all accounts that start with a value between `30` and `33` and are exactly 15 digits long
`30-33:15-18`	Match all accounts that start with a value between `30` and `33` and are between 15 and 18 digits long
`30-33:15-18:DISC`	Match only Discover cards that start with a value between `30` and `33` and are between 15 and 18 digits long
`0-9::VISA:2`	Match only Visa Purchase cards
`4;30-33:15-18:DISC`	Match any account that starts with `4` and Discover cards that start with a value between `30` and `33` and are between 15 and 18 digits long
`0-9::VISA:2;0-9::MC:2`	Match only Visa and Mastercard Purchase cards