Introduction

The electronic check transaction commands detailed in this API give merchants a comprehensive platform for authorizing, modifying, voiding and processing electronic check payments   on the ACHQ platform through their own software.

The purpose of this document is not only to explain how to submit an ACH payment for authorization and processing, but also to present the concepts necessary for understanding what data is required, why, and what the response data should mean to your software application and your customers. Please read each section carefully.

An Overview of Electronic Check Processing

Electronic checks, also referred to as ACH (Automated Clearing House) transactions, are check transactions processed electronically through the Federal Reserve. Merchants may utilize the ACH system to initiate electronic check payments to and from their customers, employees, vendors, etc.

The ACH system is self-regulated by the banking industry through an association of banks called NACHA (the National Automated Clearing House Association). NACHA works in conjunction with the Federal Reserve to establish the rules and regulations that govern all aspects of exchanging these electronic checks between the banks and credit unions.

Unlike credit cards, electronic check transactions cannot be instantly verified and approved. This is due to the rules that govern the United States Federal Reserve and the autonomous nature of each bank. The banking industry is working to make all forms of check processing and authorization occur in real-time, but due to political, economic, and consumer protection issues, that technology is still years away from widespread implementation.

What has been implemented instead is a system that still provides authorization, but over a period of days instead of seconds. There are several reasons for the slowness of the process, but the bottom line is that the process has been made logistically slow on purpose to protect bank account owners from fraud and error.

Instead of being sent between banks in real-time, all electronic check transactions are batched and sent between banks overnight through the Federal Reserve on banking business days only. A bank can take up to four business days to indicate if a transaction is rejected (Returned) and why. If no response is received within that time frame, the transaction is considered approved (Cleared).

Please note that although banks are expected to respond with a Return within the four business day time frame, there are some exceptions to this rule because of laws that protect the customer. If a Return is received after a check payment has been Cleared and funds have been settled to the merchant's account, the Return transaction is classified as a Charged Back because funds will need to be "charged back" from the merchant's account and returned to the customer's bank.

Standard Entry Class (SEC) Codes

NACHA regulations require that a transaction submitted to the Federal Reserve for processing must include something called a Standard Entry Class (SEC) Code to communicate exactly how the customer gave the merchant authorization to debit/credit their bank account.

The following table shows the proper SEC Codes to use depending on how the merchant obtained authorization to debit or credit an individual or company's bank account:

Available Standard Entry Class (SEC) Codes

Authorization Method	SEC Code
Document Signed by Individual	PPD
Document Signed by Company	CCD
Via the Internet	WEB
Recorded Telephone Call	TEL
Non-Recorded Telephone Call with Notification	TEL

If NACHA decides to audit a merchant or processor, part of the audit process may require providing proof of the authorization method (SEC Code) specified for any given transaction. Failure to properly comply and provide proof of the authorization can result in fines up to $10,000 for each transaction in violation, so it is important that you correctly indicate the SEC Code and maintain good records of your authorizations. Additional information about SEC Codes can be provided upon request.

The ACHQ Gateway and the Command/Response System

The ACHQ gateway supports the ability to perform a variety of ACH payment management tasks including:

- Creating and authorizing new payment transactions
- Modifying or canceling existing payments
- Uploading or canceling batches of payments
- Querying payment data for status updates
- Retrieving reports

This document details the specific subset of ACHQ Gateway commands dedicated to creating, modifying, voiding or verifying individual ACH payments in real-time.

Every task performed through the ACHQ Gateway is achieved through a command/response method. Simply put, your software issues a specific command to the ACHQ Gateway to accomplish your desired task and the gateway will send back a formatted response to your command indicating the success or failure of that task along with any information necessary to resolve or understand why a command may have failed.

Available ECheck Transaction Commands

The following is a list and brief explanation of the ECheck Transaction Commands that can be issued through the *ACHQ HTTP POST API:*

- **ECheck.ProcessPayment** – Initiates an ACH payment to or from a bank account. Validates the transaction data packet for errors, performs basic routing and account number validation and optionally verifies the bank account using the *Express Verify* service. Payment is only processed to the Federal Reserve if it passes the validation process.

- **ECheck.VerifyBankAccount** – Performs basic algorithm verification on routing and account numbers and optionally verifies the bank account using the ACHQ *Express Verify* service.

- **ECheck.CreateACHQToken** – Tokenizes the bank account information provided. Validates the transaction data packet for errors, performs basic routing and account number validation.

- **ECheck.Update** – Modifies a transaction if it has not yet been sent to the Federal Reserve for processing at the daily cut-off time.

- **ECheck.Refund** – Reverses a previous debit that has settled (Cleared) and refunds the same amount (or less if specified) to the same bank account as the original referenced debit. Refunds are automatically captured for settlement.

- **ECheck.Void** – Cancels (stops) an ECheck payment from processing if it has not yet been sent to the Federal Reserve for processing.

- **ECheck.Hold** – Places a Scheduled transaction on indefinite 'Merchant Hold' status if it has not yet been sent to the Federal Reserve for processing. Authorized transactions cannot be placed on hold.

- **ECheck.RemoveHold** – Removes the 'Merchant Hold' status on a transaction and reschedules the transaction for processing on the date specified in the command.

Verification Using *Express Verify*

Merchants may choose to sign up for an optional bank account verification service called *Express Verify*. This service can report in real-time whether an account exists or whether it is currently overdrawn, frozen or closed thus ascertaining whether a check is likely to be returned.

The service returns a 3-letter verification status which can be "POS" (positive) indicating the bank account is found and in good standing, "NEG" (negative) indicating the account does not exist or is not in good standing, and "UNK" (unknown) indicating the bank account does not belong to a participating bank. The code "ERR" (error) can also result if technical problems occurred verifying the account.

Please see *Appendix B – Express Verify Response Codes* for a complete list of possible responses from the *Express Verify* system and their meanings.

Unique Payment Transaction Identification

Proper communication between two separate transaction management applications (like this gateway and your software application) requires that both applications share a common, unique reference for each transaction for the two applications to communicate intelligently.

The ACHQ Gateway supports the following methods for uniquely referencing transactions:

Transact_ReferenceID

At the completion of each successful *ECheck.ProcessPayment* or *ECheck.Refund* command, a *Transact_ReferenceID* value is returned by the gateway. This value must be used to update or void a pending transaction. The *Transact_ReferenceID* value is also included in all transaction reports as a tool for cross-referencing your internal transactions with the report results. Your software will need to store this value and associate it internally with your own payment transaction record for later transaction cross-reference.

Merchant_ReferenceID

The ACHQ Gateway also supports the ability for Merchants to assign their own unique internal ID's to each payment transaction. Although this value cannot be used to update or void pending transactions, the *Merchant_ReferenceID* value will be included in all transaction reports as an alternative (and easier) method of cross-referencing your internal transactions with the report results.

Tokenization

Tokens are randomized strings of information that can be used to represent a bank account. The security benefits are obvious. If your company does not have to store actual bank account information, this reduces your risk of exposing the private banking information of your customers to potential hackers or even malicious employees.

When initiating an ACH payment, you have the option of supplying either a bank routing and account number or a supported token that represents a bank account.

ACHQ provides tokenization of bank accounts as an optional service for your benefit and security. When creating an ACH payment for the first time through the *Echeck.ProcessPayment* command, you have the option to tokenize your customer's bank account information through a parameter called *Create_ACHQToken.* The ACHQ token will be sent back in the command response object. Once created, you can store and use that token for all future ACH payments to or from that person or company.

ACHQ also supports tokens from certain tokenization providers. A tokenization provider is a company that either manages tokens on your behalf or provides tokens that represent bank accounts as a part of their banking services.

If you are supplying an ACHQ token or a token from a service provider that has integrated or partnered with ACHQ, you simply supply the token in the *AccountToken* field of the appropriate payment command and ACHQ will convert that token to the actual bank account for ACH payment processing.

If you do not have a token, then you obviously will need to provide a valid bank routing and account number to process a payment. Providing an *AccountToken* and a bank account in the same payment command will result in an error.

Application Testing

The ACHQ SpeedChex API supports sandbox, development and production modes through a single API endpoint. Please note the following methods for differentiating and controlling your integration mode for the ACHQ SpeedChex API:

- **Sandbox Mode** - ACH payments are sent to a Sandbox account where they are captured, stored and show up in status tracking reports and queries - but they are never sent to the Federal Reserve for processing. You must use the following credentials for sending payments to the Sandbox account

- **Development Mode** - In Development mode, ACH payments are submitted to the ACHQ API using your production credentials with a *TestMode* parameter set to "On". Payments submitted in this mode will receive a valid success or failure response, but unlike Sandbox and Production modes - these ACH payments are never stored on the ACHQ platform for processing or made available for reporting and status tracking queries.

The following sandbox credentials and bank account rules may be helpful for testing your integration:

Sandox Provider API Credentials
ProviderID: 99
Provider_GateID: test
Provider_GateKey: test

Sandbox Merchant API Credentials
MerchantID: 2001
Merchant_GateID: test
Merchant_GateKey: test

Testing Bank Accounts	Sandbox Testing Scenarios
Routing: 123123123 Account: \<any account # not on this list\>	Passes all bank account verifications. Transaction status is marked *Cleared* four business days after its created
Routing: 123123123 Account: 111111111	Transaction status marked as *Returned NSF* on the next banking business day
Routing: 123123123 Account: 222222222	Transaction status marked as *Returned Other* (administrative return) on the next banking business day
Routing: 123123123 Account: 444444444	Transaction status marked *Cleared* 4 business days after its created and then it gets marked to *Charged Back* on the next business day
Routing: 123123123 Account: 987654321	Transaction fails *Express Verify* and is immediately rejected

Please Note: Merchant ID 2001 is a public test account. Any information you transmit may be viewed by users running the ACHQ demo. This includes payment accounts, names, addresses, phone numbers, and email addresses. Please always use fabricated test information.

Data Security and Protection

Every Merchant is assigned a unique Provider ID, GateID, and GateKey that must be kept confidential and will be required as part of each data packet sent to the ACHQ Gateway. In addition, an IP filtering scheme may be implemented to ensure that command packets are only accepted from IP addresses registered by the Merchant.

Protecting the financial transaction data processed through the ACHQ Gateway is of utmost priority. This means not only implementing the highest levels of security standards in data encryption and system security, but also setting strict controls that limit authorized access to sensitive information.

Overview of the Command Process

Integrating this API into your software application is not difficult. The following is an overview of the major components of this task:

- **Data Gathering** - Merchants are responsible for collecting and submitting all data associated with a transaction command.

- **Submitting a Gateway Command** – You must create a script or program that submits a command packet over the Internet using the secured HTTPS protocol. The rules for constructing the HTTPS command are defined in the next section of this document titled *General Implementation Rules and Specifications*.

- **Transaction Packet Validation** – The ACHQ Payment Gateway parses the command packet for missing data, invalid field values, security violations, etc.

- **Response Processing** - The ACHQ Payment Gateway will return a comma-delimited or JSON text response after it receives and processes the command. The exact format and meaning of the text response will be based on the command issued as defined in the next section of this document titled *General Implementation Rules and Specifications*.

General Implementation Rules and Specifications

1. **Basic Command Template** – The following table defines the basic data fields that will be required when submitting any command through the *ACHQ HTTP POST API:*

Field Name	Usage	Field Value Format Constraints
MerchantID	Required	Provided by your ACHQ account manager
Merchant_GateID	Required	Provided by your ACHQ account manager
Merchant_GateKey	Required	Provided by your ACHQ account manager
Command	Required	Set this value to the name of the command you are issuing to the ACHQ Payment Gateway.
CommandVersion	Required	Set to **2.0** to enforce this API documentation revision.
TestMode	Optional	Set this value to 'On' to test a command response without actually executing the command. Default value is 'Off'.
ResponseType	Optional	Value must be: **JSON**, or **CSV**. Default is **JSON**
<additional fields as required>		Based on the Command value, you may be required to define additional fields to send in the command packet. These fields will be defined in the various sections of this document dedicated to each specific command.

2. **Submitting a Gateway Command** – To submit a gateway command to ACHQ, use either HTTP POST or GET to transmit the necessary command fields to the following secure URL:

> https://www.speedchex.com/datalinks/transact.aspx

If using HTTP GET, please make sure that all data values have been encoded to be URL-safe.

3. **Response Template** – In response to any command, the ACHQ Payment Gateway will return a simple comma-delimited text response indicating whether the command succeeded or failed and any additional information necessary to interpret the response. The following table defines the field values that will be returned in every response and their meaning:

Field Name	Field Contents	Max Length	Additional Information
CommandStatus	Returns one of the following values: • *Approved* • *Declined* • *Error*	30	Indicates the success or failure of the command issued.
ResponseCode	A 3 digit code indicating command success or reason for command failure.	3	Please refer to *Appendix A - Response Code Definitions* for a list of possible ResponseCode values, their descriptions, and what ErrorInformation may be made available.
Description	Description of the *ResponseCode* value	255	
ErrorInformation	Additional information to help determine the source of an error.	50	
ExpressVerify.Status	Returns one of the following values if *Express Verify* is activated: 'POS', 'NEG', 'UNK' or 'ERR'.	3	ExpressVerify is a complex object that contains the bank account verification results from *Express Verify*.
ExpressVerify.Code	A code indicating the reason for the *ExpressVerify*.Status value	5	
ExpressVerify.Description	A brief explanation for the *ExpressVerify*.Status value	255	Please refer to **Appendix B - Express Verify Response Codes** for more details about these responses.
ResponseData	Please see the documentation for the specific command to be issued for an explanation of the possible value(s) for this field.		This is a generic object that can take the form of any scalar or complex object called for by the command that is issued.
ACHQToken	A unique token that can be used to represent the supplied bank routing and account information on any future payments.	50	A token will only be generated if directed requested on the *Create_ACHQToken* parameter.
Provider_TransactionID	This field is returned but not used by this API	50	This value will always be null
Transact_ReferenceID	A unique ID assigned to each command submitted to the ACHQ Gateway.	30	This value can be used for as a unique transaction identifier or as a reference for support on any command.

Command: ECheck.ProcessPayment

Description: Initiates an ACH payment to or from a bank account. Validates the command parameters for errors, performs basic routing and account number validation and optionally verifies the bank account using the *Express Verify* service. Payment is only processed to the Federal Reserve if it passes the validation process.

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
Command	Required	Set to **ECheck.ProcessPayment**	50
CommandVersion	Required	Set to **2.0** for this API documentation revision.	-
TestMode	Optional	Set this value to **On** to test a command response. Default is **Off**.	3
DateScheduled	Optional	Date to process this payment. Format mm/dd/yyyy	-
PaymentDirection	Required	Value must be **FromCustomer** or **ToCustomer**	12
Amount	Required	The amount of the check.	-
LocationName	Optional	Any location predefined by the merchant in their administration console	50
Merchant_ReferenceID	Optional	A unique internal identifier you have assigned to this payment	50
Description	Optional	A description for this transaction	100
Billing_CustomerID	Optional	A unique ID assigned to the Customer.	20
Billing_CustomerName	Required	Name of person on bank account	80
Billing_Company	Conditional	Company name on bank account. Required if *CheckType* field is set to **Business**	80
Billing_Address1	Required	Street Address on bank account	70
Billing_Address2	Optional	Additional street address information	40
Billing_City	Required	City	70
Billing_State	Required	State	30
Billing_Zip	Required	Zip Code. Format: ##### or #####-####	10
Billing_Country	Optional	2-letter country code (ISO 3166). Default is **US**	2
Billing_Phone	Required	Phone number	20
Billing_Email	Conditional	Payment notification email address. Required if *SendEmailToCustomer* is set to *Yes*	80
SendEmailToCustomer	Required	Value must be **Yes** or **No**	3
Customer_IPAddress	Conditional	Customer's IP Address. Only required if the *SECCode* field is set to **WEB**.	15
Run_ExpressVerify	Required	Value must be **Yes** or **No**	3
Create_ACHQToken	Optional	Value must be Yes or No. Default is No if not specified.	3
SECCode	Required	Value must be **PPD**, **CCD**, **WEB**, or **TEL**.	3
ResponseType	Optional	Must be **JSON** or **CSV**. Default is **JSON**	4
Only Required When Submitting a Tokenized Bank Account			
AccountToken	Required	The token representing a customer's bank account information	100
TokenSource	Required	Possible values: **ACHQ** or **Plaid**	50
Only Required When Submitting Standard Bank Account Information			
CheckType	Required	Value must be **Personal** or **Business**	8
AccountType	Required	Value must be **Checking** or **Savings**	8
RoutingNumber	Required	ABA routing number on customer's check.	9
AccountNumber	Required	Bank account number to debit or credit	30
CheckNumber	Optional	Customer's check number.	25

Command: **ECheck.VerifyBankAccount**

Description: Performs basic algorithm verification on a routing number and account number and optionally verifies the bank account using the Express Verify service. The following table defines the data field rules for this command:

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
CommandVersion	Required	Set to **2.0** for this API documentation revision	-
TestMode	Optional	Set this value to **On** to test a command response. Default value is **Off**.	3
PaymentDirection	Required	Value must be **FromCustomer** or **ToCustomer**	12
Amount	Required	The amount of the check	-
CheckType	Required	Value must be **Personal** or **Business**	8
AccountType	Required	Value must be **Checking** or **Savings**	8
RoutingNumber	Required	ABA routing number on customer's check	9
AccountNumber	Required	Customer's bank account number	30
Run_ExpressVerify	Required	Value must be **Yes** or **No**	3
ResponseType	Optional	Must be **JSON** or **CSV**. Default is **JSON**	4

Response Note:

If *Run_ExpressVerify* is set to **No**, this command will perform a simple verification to make sure the bank routing and account numbers pass a basic algorithm structure test. A failure will result in a ResponseCode value of 209 (Invalid Routing Number) or 210 (Invalid Bank Account Number).

If *Run_ExpressVerify* is set to **Yes** and the merchant account is setup on the ACHQ Express Verify service, the API will connect to the banking network to validate the bank routing and account numbers exist and have a good account status.

If Express Verify fails, the API will return a failure *ResponseCode* value of 202 and an E*xpressVerify.Status* value of **NEG** (negative).

If Express Verify succeeds or the account status cannot be determined, the API will return a *ResponseCode* value of **000** (Command Successful. Approved) and the *ExpressVerify.Status* response value will be set to **POS** (positive) or **UNK** (unknown).

Please refer to **Appendix B - Express Verify Response Codes** for more details about the possible responses from the Express Verify service to make sure your code handles each type of response appropriately according to your company's or your merchant's policies.

Command: ECheck.CreateACHQToken

Description: Tokenizes the bank account information provided.   Validates the transaction data packet for errors, performs basic routing and account number validation. The following table defines the data field rules for this command:

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
Command	Required	Set to **ECheck.CreateACHQToken**	50
CommandVersion	Required	Set to **2.0** for this API documentation revision	-
CheckType	Required	Value must be **Personal** or **Business**	8
AccountType	Required	Value must be **Checking** or **Savings**	8
RoutingNumber	Required	ABA routing number on customer's check	9
AccountNumber	Required	Customer's bank account number	30
ResponseType	Optional	Must be **JSON** or **CSV**. Default is **JSON**	4

Command: ECheck.Update

Description: Modifies a transaction if it has not yet been sent to the Federal Reserve for processing.

Note: Please do not specify a value for an optional field if you do not intend to modify its value.

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
Command	Required	Set to **ECheck.Update**	50
CommandVersion	Required	Set to **2.0** for this API documentation revision.	-
TestMode	Optional	Set this value to **On** to test a command response. Default value is **Off**.	3
Transact_ReferenceID	Required	The unique *Transact_ReferenceID* sent in response to the original *ECheck.ProcessPayment* command.	50
DateScheduled	Optional	Date to process this payment. Format mm/dd/yyyy	-
PaymentDirection	Optional	Value must be **FromCustomer** or **ToCustomer**	12
Amount	Optional	The amount of the check.	-
CheckType	Optional	Value must be **Personal** or **Business**	8
AccountType	Optional	Value must be **Checking** or **Savings**	8
RoutingNumber	Optional	ABA routing number on customer's check.	9
AccountNumber	Optional	Customer's bank account number	30
CheckNumber	Optional	Customer's check number. Only required for non-ACH merchants.	25
LocationName	Optional	Any location pre-defined by the merchant in their administration console	50
Merchant_ReferenceID	Optional	A unique internal identifier you have assigned to this payment	50
Description	Optional	A description for this transaction	100
Billing_CustomerID	Optional	A unique ID assigned to the Customer.	20
Billing_CustomerName	Required	Name of person on bank account	80
Billing_Company	Conditional	Company name on bank account. Required if CheckType field is set to **Business**	80
Billing_Address1	Required	Street Address on bank account	70
Billing_Address2	Optional	Additional street address information	40
Billing_City	Required	City	70
Billing_State	Required	State	30
Billing_Zip	Required	Zip Code (format: ##### or #####-####)	10
Billing_Country	Optional	2-letter country code (ISO 3166). Default is **US**	2
Billing_Phone	Required	Phone number	20
Billing_Email	Conditional	Payment notification email address. Required if SendEmailToCustomer is set to **Yes**	80
SendEmailToCustomer	Optional	Value must be **Yes** or **No**	3
Customer_IPAddress	Optional	Payer's IP Address. Only required if the SECCode field is set to **WEB**.	15
Run_ExpressVerify	Optional	Value must be **Yes** or **No**	3
Create_ACHQToken	Optional	Value must be Yes or No. Default is No if not specified.	3
SECCode	Optional	Value must be **PPD**, **CCD**, **WEB**, or **TEL**.	3
ResponseType	Optional	Must be **JSON** or **CSV**. Default is **JSON**	4

Command: ECheck.Refund

Description: Reverses a previous debit that has settled (Cleared) and refunds the same amount (or less if specified) to the same bank account as the original referenced debit. Refunds are automatically captured for settlement. The following table defines the data field rules for this command:

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
Command	Required	Set to **ECheck.Refund**	-
CommandVersion	Required	Set to **2.0** for this API documentation revision	-
TestMode	Optional	Set this value to **On** to test a command response. Default value is **Off**.	3
Transact_ReferenceID	Required	The unique Transact_ReferenceID sent in response to the original *ECheck.ProcessPayment* command.	50
DateScheduled	Optional	Date to process this payment. Format mm/dd/yyyy	-
Amount	Optional	Use this field if you need to specify a refund amount less than the original debit amount.	-
Merchant_ReferenceID	Optional	A unique internal identifier you have assigned to this payment	50
Description	Optional	A description for the refund transaction	100
ResponseType	Optional	Must be **JSON** or **CSV**. Default is **JSON**	4

Command: Echeck.Void

Description: Cancels (stops) a transaction from processing to the Federal Reserve. The following table defines the data field rules for this command:

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
Command	Required	Set to **ECheck.Void**	50
CommandVersion	Required	Set to 2.0 for this API documentation revision.	-
TestMode	Optional	Set this value to **On** to test a command response. Default value is **Off**.	3
Transact_ReferenceID	Required	The unique Transact_ReferenceID sent in response to the original *ECheck.ProcessPayment* command.	50
ResponseType	Optional	Must be **JSON** or **CSV**. Default is **JSON**	4

Command: ECheck.Hold

Description: Places a scheduled transaction on 'Merchant Hold' if it has not yet been sent to the Federal Reserve for processing. This is a delay in processing that a merchant may impose on a transaction until they are ready to remove the hold. Authorized transactions cannot be placed on hold. The following table defines the data field rules for this command:

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
Command	Required	Set to **ECheck.Hold**	50
CommandVersion	Required	Set to **1.0** for this API documentation revision.	-
TestMode	Optional	Set this value to **On** to test a command response. Default value is **Off**.	3
Transact_ReferenceID	Required	The unique Transact_ReferenceID sent in response to the original *ECheck.ProcessPayment* or *ECheck.Refund* command.	50
ResponseType	Optional	Must be **JSON** or **CSV**. Default is **JSON**	4

Command: ECheck.RemoveHold

Description: Removes the 'Merchant Hold' status on a transaction and reschedules the transaction for processing on the date specified in the command. The following table defines the data field rules for this command:

Field Name	Usage	Field Value Format Constraints	Max Length
MerchantID	Required	Provided by your ACHQ account manager	-
Merchant_GateID	Required	Provided by your ACHQ account manager	-
Merchant_GateKey	Required	Provided by your ACHQ account manager	-
Command	Required	Set to **ECheck.RemoveHold**	50
CommandVersion	Required	Set to **2.0** for this API documentation revision.	-
TestMode	Optional	Set this value to **On** to test a command response. Default value is **Off**.	3
Transact_ReferenceID	Required	The unique Transact_ReferenceID sent in response to the original *ECheck.ProcessPayment* or *ECheck.Refund* command.	50
DateScheduled	Required	Date to process this payment. Format mm/dd/yyyy	-
ResponseType	Optional	Must be **JSON**, or **CSV**. Default is **JSON**	4

Appendix A – Response Code Definitions

Response Code	Description	Contents of the ErrorInformation Field
GATEWAY COMMAND SUCCESS		
000	Command Successful. Approved.	
GATEWAY COMMAND ERRORS		
100	Invalid Gateway Credentials	Credential Object Name
101	Invalid Gateway Command	
102	Duplicate Command Not Processed	Transact_ReferenceID of the original Command
103	Transaction Cannot Be Modified	Transaction Status
104	Batch Cannot Be Modified	Batch Status
105	Invalid Transact_ReferenceID	
106	Invalid BatchID	
107	Non-Unique Reference/Transaction ID	Field Name
108	Invalid Reference/Transaction ID	Field Name
109	Invalid Source IP	
110	Invalid Value In Message	
INPUT DATA VALIDATION ERRORS		
150	Required Field Missing	Field Name
151	Field Value Is Not Valid	Field Name
152	Field Value Exceeds Maximum Length	Field Name
PAYMENT ACCOUNT VERIFICATION FAILURES		
200	Failed AVS	
201	Failed CVN	
202	Failed Express Verify	
203	Invalid Credit Card Number	
204	No Such Card Issuer	
205	Expired Card	
206	Invalid Expiration Date	
208	Call Issuer for Further Information	
209	Invalid Routing Number	
210	Invalid Bank Account Number	
211	Invalid PIN	
212	Invalid ACHQToken	
PAYMENT ACCOUNT DECLINES		
300	Transaction was Declined by Processor	
301	Transaction was Rejected by Gateway	
302	No Card Number on File with Issuer	
304	Invalid Account Type	
305	Account Closed	
306	Account Inactive	

Appendix A – Response Code Definitions

Response Code	Description	Contents of the ErrorInformation Field
PAYMENT ACCOUNT DECLINES (*continued...*)		
307	Account Frozen	
309	Insufficient Funds	
310	Over Limit	
311	Do Not Honor	
312	Transaction Not Allowed	Reason (if known)
313	Invalid for Debit	Reason (if known)
314	Invalid for Credit	Reason (if known)
315	Customer Opt Out	Reason (if known)
316	Customer Advises Not Authorized	
317	Manual Key Not Allowed	
318	Duplicate Transaction at Processor	
319	Token Authentication Failed	
FRAUD DECLINES		
400	Pick Up Card	
401	Lost Card	
402	Stolen Card	
403	Fraudulent Card	
404	Excessive Declines From Same Source	
405	Excessive PIN Attempts	
406	Excessive Purchase Frequency	
MERCHANT DIRECTIVES FROM PROCESSOR		
500	Declined - Stop All Recurring Payments	
501	Declined - Update Cardholder Data Available	
502	Declined - Further Instructions Available	Instructions
503	Declined - Call Processor for Voice Authorization	
504	Declined - Call Processor for Fraud Instructions	
PROCESSOR ADMINISTRATIVE ERRORS		
600	Internal Gateway Error	
601	Internal Processor Error	
602	Communication Error with Issuer	
603	Communication Error with Processor	
604	Processor Feature Not Available	
605	Processor Format Error	
606	Invalid Terminal Number	
607	Merchant Not Setup	
608	Merchant Account is Inactive	
609	Invalid Merchant Configuration	
610	Invalid Payment Method for Merchant	
611	Unsupported Card Type	
OTHER		
999	Contact Support Representative	

Appendix B – Express Verify Response Codes

Bank accounts that are found and in good standing are approved with a POS (positive) result status. Accounts that do not exist or are in a negative standing at their bank are rejected with a NEG (negative) result status. If a bank account is from a bank that is not part of the *Express Verify* network or the bank is not reporting information about the bank account, the transaction is approved with an UNK (unknown) result status.

The following table shows all possible responses from the *Express Verify* system:

Status	Code	Description
ERR	E01	EXPRESS VERIFY SERVICE NOT ACTIVATED
NEG	P00	ACCT NOT LOCATED
NEG	P01	ACCOUNT CLOSED
NEG	P03	NO DEBITS
NEG	P04	NO CHECKS
NEG	P05	NSF
NEG	P06	UNCOLLECTED FUNDS
NEG	P12	ISSUER DECLINED
UNK	P40	NO INFO
NEG	P41	NEGATIVE INFO
UNK	P50	NON PARTICIPANT
POS	P70	VALIDATED
POS	P72	VALIDATED AMOUNT
NEG	V02	ACCOUNT NOT APPROVED
NEG	V10	INVALID ROUTING NUMBER
UNK	V90	PREAUTH VENDOR UNAVAILABLE
UNK	V91	PREAUTH VENDOR ERROR