

EXACTOR TAX CALCULATION API



CONFIDENTIAL

VERSION 2014.02.26

TABLE OF CONTENTS

TABLE OF CONTENTS	2
OVERVIEW	1
Tax Request XML Interface	1
Tax Request SOAP Interface	1
API URLs	1
TAX REQUEST TRANSACTION FLOW	2
Tax Calculations (Invoice Request)	2
Sales and Credits (Commit Request)	2
Refunds (Refund Request)	3
Deletions (Delete Request)	3
DIGITAL SIGNATURE	4
EXACTOR XML SCHEMA	6
Tax Request XML	7
<i>TaxRequest</i>	7
<i>CommitRequest</i>	9
<i>RefundRequest</i>	11
<i>DeleteRequest</i>	11
Tax Response XML	12
<i>TaxResponse</i>	12
<i>InvoiceResponse</i>	12
<i>CommitResponse</i>	13
<i>RefundResponse</i>	14
<i>DeleteResponse</i>	14
<i>ErrorResponse</i>	15
Supporting Types	16
<i>Primitive Types</i>	16
<i>AddressType</i>	16
<i>TaxInfoType</i>	16
<i>Sample Invoice Tax Calculation</i>	18
<i>Sample of a combined Invoice and Commit</i>	22
<i>Sample of a Refund Request</i>	25
<i>Sample of a Delete Request</i>	26
<i>Sample of a Request by a Partner on Behalf of a Merchant</i>	27
<i>Sample of a Response by a Partner on Behalf of a Merchant</i>	27
Sample Code for Generating a Digital Signature	29
APPENDIX	32
Currency Codes	32

OVERVIEW

The Exactor Sales and Use Tax Transaction API enables client systems to process sales and use tax transactions via the Exactor web service. This API includes methods for performing tax calculations on sales orders / invoices / credit memos and optionally committing / refunding those taxes for filing and remittance purposes. This API provides two methods of integration.

- XML Interface
- SOAP Interface

The XML and SOAP interfaces each provide equivalent functionality. All requests are processed in real-time and all of the data transmitted securely using SSL. In addition, clients may optionally choose to use a digital signature to ensure that all requests originate from an authenticated merchant / user. All Exactor-registered partner systems (specific partners who transmit on behalf of multiple clients) are required to use a digital signature. See the "Digital Signature" Sections for more information.

TAX REQUEST XML INTERFACE

The Exactor Tax Request XML interface allows customers to send data in XML format over the HTTP protocol to process tax calculation requests and optionally commit or refund them to the system for tax filing purposes. The client system may simply create an XML request as defined by the Exactor Tax Request XML schema, transmit it in the payload section of an HTTP POST request to the appropriate URL, and parse the results the XML response element that is immediately returned.

TAX REQUEST SOAP INTERFACE

The Exactor SOAP interface allows customers to use the Simple Object Access Protocol (SOAP) to process tax calculation requests and optionally commit them to the Exactor System for tax filing and remittance purposes. This interface provides the same functionality as the XML interface, but allows programmers to interact with methods on objects to set/get information and invoke tax calculation, commit, and refund requests rather than constructing and transmitting the XML and parsing the responses manually. The Exactor SOAP WSDL is located at the following URL and can be used with most web toolkits/IDEs to generate objects that can be used to interact with the Exactor Web Service.

API URLS

XML Schema Definition	http://www.exactor.com/ns/ExactorTaxCalculation.xsd
XML Namespace	http://www.exactor.com/ns
XML API	https://taxrequest.exactor.com/request/xml
SOAP Endpoint	https://taxrequest.exactor.com/request/soap
SOAP WSDL	https://taxrequest.exactor.com/request/soap/?wsdl

TAX REQUEST TRANSACTION FLOW

The Exactor Tax Request API supports four basic functions:

- Calculate taxes for sales orders, invoices, credit memos, etc.
- Commit completed transactions for tax return filing and remittance purposes
- Refund transactions that were previously committed (due to returns, errors, etc.)
- Delete transactions

Each of these functions may be executed by sending a *Tax Request* element to the Exactor system. Each *Tax Request* element may contain one or more *Invoice*, *Commit*, *Refund*, or *Delete Requests* (alone or in any combination). The Exactor system will immediately return a *Tax Response* element that contains response element(s) for each of the *Invoice*, *Commit*, *Refund* or *Delete* requests in the same order that they were sent in the request. Any errors that occur will result in an Error response in replacement of the *Invoice*, *Commit*, *Refund*, or *Delete* request that failed.

TAX CALCULATIONS (INVOICE REQUEST)

A merchant may submit an invoice request to execute a tax calculation to determine the taxes associated with an order without affecting a company's actual tax obligations. This is useful in cases where a merchant wishes to inform a customer of the taxes that will be applied to an order before the actual sale is complete. The request will be processed, the tax obligations will be calculated, and the invoice will be stored in the system, but it will not be included for reporting, filing, or remittance purposes unless/until it has been committed to the system via a Commit Request. Invoice requests may contain positive and/or negative gross amounts. Most new requests will contain positive amounts for line items pertaining to products and services sold, but may also contain negative amounts to apply discounts, credits, coupons, etc.

SALES AND CREDITS (COMMIT REQUEST)

A merchant may submit a commit request to reflect that a sale or credit has been finalized thereby creating a potential tax obligation. This may be the point in time when the payment has been received (if the merchant using cash based accounting system) or when the products are shipped / services are rendered (if the merchant is using an accrual based accounting system). The Exactor system supports both methodologies and relies on the client to execute commit requests as tax obligations become due.

There are three ways that a client may submit a commit request:

- A merchant may submit a commit request that contains a transaction ID returned by prior Invoice Request (i.e. a tax calculation request that has not yet been committed). Submitting a commit request using a prior transaction ID will result in the transaction being committed as it was originally calculated.

- A merchant may submit a commit request that contains an invoice request. This will result in an atomic tax calculation and commit operation.
- A merchant may submit a commit request that contains both a prior transaction ID and an invoice request. This will result in an atomic replacement of the committed transaction represented by the prior transaction id with the data and tax calculation results of the supplied invoice request. This essentially replaces one committed transaction with a new one and is useful in cases where an invoice is modified after it has already been committed.

Once a transaction has been committed, it will be included for all reporting, filing and remittance purposes.

REFUNDS (REFUND REQUEST)

A merchant may refund a commit request by sending a refund request. This will essentially reverse (or negate) all of the taxes that were committed for the given transaction ID effective on the Refund Date. A refund request always refers to a committed invoice. Once a refund transaction has been executed, it will be included for all reporting, filing, and remittance purposes.

DELETIONS (DELETE REQUEST)

A merchant may delete any transaction by sending a delete request regardless of whether the transaction was just a pure tax calculation, committed or refunded.

DIGITAL SIGNATURE

Merchants may optionally use a digital signature to ensure that requests originate from an authentic merchant / user. Partners are required to use digital signatures when sending requests on behalf of their clients.

Merchants or partners that wish to use a digital signature should contact Exactor to obtain a "secret key". This key will be provided for a specific Merchant ID and User ID. The secret key should be kept secret to avoid compromising the authentication process.

The merchant or partner will need to implement client-side code to generate the digital signature for each tax request before transmitting requests to Exactor. This can be accomplished by generating a digest using HMAC-SHA256 together with the secret key to hash a sequence of elements within the request. The resulting value should then be passed within the DigitalSignature element of the TaxRequest. The Exactor system will use the same algorithm, secret key, and elements of the TaxRequest to generate a digest when it receives the request and then check to ensure that it matches the value passed in the DigitalSignature element. If the values do not match, the request will be rejected with an error code of "05: Missing or Invalid Digital Signature".

The client will be responsible for generating the digital signature by using these elements (in the specified order). Each element must be separated by a newline ("\n"). In order to prevent ambiguity, the element string value must replace any embedded newlines with a space. If an element is not set, the empty string ("") should be used.

- MerchantId
- UserId
- Each InvoiceRequest
 - SaleDate
 - ShipTo fields
 - ShipFrom fields
 - Each of the gross amount values for each line item (in order)
- Each CommitRequest
 - CommitDate
 - PriorTransactionId
 - InvoiceRequest (fields as listed above in the InvoiceRequest)
- Each RefundRequest
 - RefundDate
 - PriorTransactionId
- Each DeleteRequest

- PriorTransactionId

Please see the "Sample Code for Generating a Digital Signature" section of this document for some sample code (written in Ruby) for generating the digital signature.

EXACTOR XML SCHEMA

The Exactor XML Schema supports tax requests made through direct HTTP and HTTPS POST commands. The schema contains one element for requests and another for responses.

- **TaxRequest** – The *TaxRequest* element may contain one or more *InvoiceRequest*, *CommitRequest*, *RefundRequest*, or *DeleteRequest* elements.
 - **InvoiceRequest** – An invoice request is used to submit an invoice into the system to calculate the tax obligations. All *InvoiceRequests* will be stored on the system, but only those that have been committed will be used for reporting, filing, and remittance purposes.
 - **CommitRequest** – A commit request is used to commit a new *InvoiceRequest* or a prior *InvoiceRequest* that has not yet been committed.
 - **RefundRequest** – A refund request is used to refund a committed *InvoiceRequest*.
 - **DeleteRequest** – A delete request is used to delete any transaction.
- **TaxResponse** – The *TaxResponse* element will contain a response for every *InvoiceRequest*, *CommitRequest*, *RefundRequest*, or *DeleteRequest* passed in the *TaxRequest*. The responses will be in the same order that they were received. Each *InvoiceRequest*, *CommitRequest*, *RefundRequest*, and *DeleteRequest* will either have a corresponding *InvoiceResponse*, *CommitResponse*, *RefundResponse*, *DeleteResponse*, or *ErrorResponse* element returned.
 - **InvoiceResponse** – A response to an *InvoiceRequest*.
 - **CommitResponse** – A response to a *CommitRequest*.
 - **RefundResponse** – A response to a *RefundRequest*.
 - **DeleteResponse** – A response to a *DeleteRequest*.
 - **ErrorResponse** – A response to an *InvoiceRequest*, *CommitRequest*, *RefundRequest*, or *DeleteRequest* that could not be processed due to errors.

TAX REQUEST XML

TaxRequest

Field Name	Data Type	Required	Max #	MaxSize	Description
<TaxRequest>	TaxRequestType	Yes	1	n/a	A tax request. This must contain one or more InvoiceRequest, CommitRequest, RefundRequest, or DeleteRequest elements. This field has two optional attributes: version – Is used to pass in the version of the Schema. plugin – Is used to pass in the name of the plugin connecting to Exactor.
<MerchantId>	String	Yes	1	8	An Exactor issued merchant ID. This is an 8 digit number assigned by Exactor to each merchant account.
<UserId>	String	Yes	1	12	A user ID for the merchant or partner account. This is a name that is 6-20 characters in length.
<PartnerId>	String	No	1	8	An Exactor issued partner ID. This is an 8 digit number assigned by Exactor to each partner account.
<DigitalSignature>	Base64Binary	No	1	n/a	A digital signature.
<InvoiceRequest>	InvoiceRequestType	No	∞	n/a	Invoice request(s) to create new invoices.
<CommitRequest>	CommitRequestType	No	∞	n/a	Requests to commit new or previously created invoices for reporting, filling, and remittance purposes.
<RefundRequest>	RefundRequestType	No	∞	n/a	Requests to refund previously committed invoices for reporting, filing, and remittance purposes.
<DeleteRequest>	DeleteRequestType	No	∞	n/a	Requests to delete any transaction.

InvoiceRequest

Field Name	Data Type	Required	Max #	Max Size	Description
<InvoiceRequest>	InvoiceRequest Type	Yes	1	n/a	The invoice request.
<SaleDate>	Date	Yes	1	10	The date of the transaction in YYYY-MM-DD format. This is the date that will be used for the tax calculation for all line items. If the line items are refunds, the sale date should be set to the same date as the original sale.
<PurchaseOrderNumber>	String	No	1	32	The Purchase Order Number.
<CurrencyCode>	CurrencyCodeType	No	1	3	The currency code associated with the invoice (see the Currency Code table). USD will be used as the default if no currency code is specified.
<TaxClass>	TaxClassType	No	1		The tax class. This can be set to "Sales" or "Use". The default is "Sales".
<TaxDirection>	TaxDirectionType	No	1	7	The direction of the tax calculation. This can be set to "Forward" (which is the default value) or "Reverse". A forward calculation will compute the taxes based on the gross amount. A reverse calculation will compute the taxes assuming that the gross amount already includes the taxes. This setting may be overridden for specific line items. This setting also requires the merchant account to be signed up for forward/reverse calculation control services.
<ExemptionId>	String	No	1	12	This can be either an Exactor-issued exemption id (ECN-00000000) or a merchant specified Customer Id that assigned to an Exemption Id. If this field contains an Exemption ID (ECN), the Exactor system will attempt to apply an entity exemption if the exemption criteria matches the invoice information (i.e. ship to state, effective dates, etc.). If this field contains a merchant specified Customer Id, the Exactor system will look up all of the Exemptions that match the Customer Id and attempt to apply the first exemption where the exemption criteria matches the invoice information.
<BillTo>	AddressType	No	1	n/a	The billing address of the invoice. This may be overridden for specific line items.
<ShipTo>	AddressType	No	1	n/a	The address where all items are shipped to. For over the counter transactions, this will be the same as ShipFrom. This may be overridden for specific line items. If the ship to address is not specified, Exactor will use the Bill To Address as the Ship To Address. If there is no Bill To Address, Exactor will use the Ship From Address as the Ship To Address (i.e. over the counter sales).
<ShipFrom>	AddressType	No	1	n/a	The address where all items are shipped from. This may be overridden for specific line items. If the Ship From Address is not specified, Exactor will use the Account address as the ship from Address.
<LineItem>	n/a	Yes	∞	n/a	The line item(s) for the request. Each line item has one required attribute: <ul style="list-style-type: none"> ▪ id="_#" (i.e. id="_1", id="_2", etc.)
<SKU>	String	No	1	16	An SKU code for the line item. This may be an Exactor SKU code (see the Exactor SKU table) or may be a Merchant SKU. If it is an Exactor SKU, the

					SKU should be preceded by "EUC-" (i.e. EUC-01030223 is the Exactor SKU for Routers). All Merchant SKUs should be mapped to Exactor SKUs in the Merchant mapping table.
<Description>	String	No	1	300	A description of the line item.
<Quantity>	Decimal	No	1	n/a	The quantity of the item.
<GrossAmount>	Decimal	Yes	1	n/a	The gross amount used to calculate tax for the line item. This may be a negative amount for discounts, credits, and refunds.
<TaxDirection>	TaxDirectionType	No	1	7	The direction of the tax calculation. This can be set to "Forward" (which is the default value) or "Reverse". A forward calculation will compute the taxes based on the gross amount. A reverse calculation will compute the taxes assuming that the gross amount already includes the taxes. This setting also requires the merchant account to be signed up for forward/reverse calculation control services.
<BillTo>	AddressType	No	1	n/a	The bill to address for the line item. This will override the BillTo field of the transaction if both are set.
<ShipTo>	AddressType	No	1	n/a	The ship to address for the line item. This will override the ShipTo field of the transaction if both are set.
<ShipFrom>	AddressType	No	1	n/a	The ship from address for the line item. This will override the ShipFrom address of the transaction if both are set.

CommitRequest

Field Name	Data Type	Required	Max #	Max Size	Description
<CommitRequest>	CommitRequest Type	Yes	1	n/a	The invoice commit request.
<CommitDate>	Date	Yes	1	10	The date that the invoice action should be committed in YYYY-MM-DD format. The commit date must follow several rules: <ul style="list-style-type: none"> The commit date must occur on or after the sale date. The commit date may be set to a date in the future (a date that has not occurred yet). The commit date may be back dated (set to a date that has already occurred), anytime within the current month (i.e. if today is 01/31/10, the commit date could be any date greater than 01/01/10). The commit date may be back dated to anytime within the previous month within the first 5 days of the current month (i.e. if today is 02/05/10, the commit date could be any date greater than 01/01/10. But if today is 02/06/10, the commit date must be greater than 02/01/10). This allows merchants to close out books within the first 5 days of the following month.
<InvoiceNumber>	String	Yes	1	32	The merchant supplied invoice number for the invoice to be committed. This can be any data that the

					merchant would like to store, and is not used by Exactor in any way.
<InvoiceRequest>	InvoiceRequestType	No	1	n/a	An invoice request. This will cause a new calculation to occur (using the information in this element) regardless of whether a PriorTransactionId is specified.
<PriorTransactionId>	String	No	1	36	The transaction id of the invoice request to be committed. This can be used to commit a prior Invoice request OR to replace an existing Commit Request. You can replace an existing request by specifying the transaction id to replace in the PriorTransactionId field, and by placing the information for the new invoice into the InvoiceRequest.

RefundRequest

Field Name	Data Type	Required	Max #	Max Size	Description
<RefundRequest>	RefundRequest Type	Yes	1	n/a	The refund request.
<RefundDate>	Date	Yes	1	10	<p>The date that the invoice action should be refunded in YYYY-MM-DD format. The refund date must follow several rules:</p> <ul style="list-style-type: none"> • The refund date must occur on or after the commit date. • The refund date may be set to a date in the future (a date that has not occurred yet). • The refund date may be back dated (set to a date that has already occurred), anytime within the current month (i.e. if today is 01/31/10, the refund date could be any date greater than 01/01/10). • The refund date may be back dated to anytime within the previous month within the first 5 days of the current month (i.e. if today is 02/05/10, the refund date could be any date greater than 01/01/10. But if today is 02/06/10, the refund date must be greater than 02/01/07). This allows merchants to close out books within the first 5 days of the following month.
<PriorTransactionId>	String	Yes	1	36	The transaction id of the commit request to be refunded.

DeleteRequest

Field Name	Data Type	Required	Max #	Max Size	Description
<DeleteRequest>	DeleteRequest Type	Yes	1	n/a	The delete request.
<PriorTransactionId>	String	Yes	1	36	The transaction id of the request to be deleted.

TAX RESPONSE XML

TaxResponse

Field Name	Data Type	Required	Max #	Width	Description
<TaxResponse>	TaxResponseType	Yes	1	n/a	A response to a TaxRequest. This will contain a response for every request: - InvoiceRequest -> InvoiceResponse or ErrorResponse - CommitRequest -> CommitResponse or ErrorResponse - RefundRequest -> RefundResponse or ErrorResponse - DeleteRequest -> Delete Response or ErrorResponse
<MerchantId>	String	Yes	1	8	The merchant ID.
<UserId>	String	Yes	1	20	The user ID.
<PartnerId>	String	No	1	8	The partner ID.
<InvoiceResponse>	InvoiceResponseType	No	∞	n/a	Successful response(s) to Invoice request(s).
<CommitResponse>	CommitResponseType	No	∞	n/a	Successful response(s) to Commit request(s).
<RefundResponse>	RefundResponseType	No	∞	n/a	Successful response(s) to Refund request(s).
<DeleteResponse>	DeleteResponseType	No	∞	n/a	Successful response(s) to Delete request(s).
<ErrorResponse>	ErrorResponseType	No	∞	n/a	An error response that corresponds to an invoice, commit, refund, or delete request that failed.

InvoiceResponse

Field Name	Data Type	Required	Max #	Width	Description
<InvoiceResponse>	InvoiceResponse Type	Yes	1	n/a	The invoice response.
<TransactionId>	String	Yes	1	36	A unique transaction id assigned by the Exactor system.
<TransactionDate>	DateTime	Yes	1	n/a	The date and time that the transaction was processed by the Exactor system.
<SaleDate>	Date	Yes	1	10	The date that the sale has/will occur in YYYY-MM-DD format. This is the date used for the tax calculation.
<PurchaseOrderNumber>	String	No	1	32	The Purchase Order Number.
<CurrencyCode>	CurrencyCodeType	Yes	1	3	The currency code associated with the invoice (see the Currency Code table).
<TaxClass>	TaxClassType	Yes	1	n/a	The tax class. This will be set to Sales or Use.
<TaxDirection>	TaxDirectionType	Yes	1	7	The direction of the tax calculation. This will be set to

					"Forward" (which is the default value) or "Reverse".
<ExemptionId>	String	No	1	12	The exemption id applied to this invoice.
<GrossAmount>	Decimal	Yes	1	n/a	The gross amount of the invoice.
<TotalTaxAmount>	Decimal	Yes	1	n/a	The total tax for the invoice.
<TaxObligation>	TaxObligationType	Yes	1	n/a	The tax obligation of the merchant for this invoice. <ul style="list-style-type: none"> • Full – All items in the invoice are subject to taxation based on the registered service areas. • Partial – One or more items (but not all) in the invoice are subject to taxation based on the registered service areas. • None – None of the items in the invoice are subject to taxation based on the registered service areas.
<LineItem>	n/a	Yes	∞	n/a	The line item(s) from the calculation request. Each line item has one required attribute: <ul style="list-style-type: none"> ▪ id="_{#}" (i.e. id="_1", id="_2", etc.)
<GrossAmount>	Decimal	Yes	1	n/a	The gross amount of the line item.
<TaxDirection>	TaxDirectionType	Yes	1	7	The direction of the tax calculation. This will be set to "Forward" (which is the default value) or "Reverse".
<TotalTaxAmount>	Decimal	Yes	1	n/a	The total tax amount for the line item.
<TaxInfo>	TaxInfoType	No	∞	n/a	Details about each of the taxes applied to the line item. This element is only returned for merchants who sign up to receive advanced tax details with their service plan.

CommitResponse

Field Name	Data Type	Required	Max #	Width	Description
<CommitResponse>	CommitResponse Type	Yes	1	n/a	The commit response.
<TransactionId>	String	Yes	1	36	A unique transaction id assigned by the Exactor system.
<TransactionDate>	DateTime	Yes	1	n/a	The date and time that the transaction was processed by the Exactor system.
<CommitDate>	Date	Yes	1	10	The commit date for this invoice transaction in YYYY-MM-DD format.
<InvoiceNumber>	String	Yes	1	32	The merchant supplied invoice number.
<InvoiceResponse>	InvoiceResponse Type	No	1	n/a	The invoice (if embedded in the request).
<PriorTransactionId>	String	No	1	36	The transaction id that this commit request applies to.

RefundResponse

Field Name	Data Type	Required	Max #	Width	Description
<RefundResponse>	RefundResponse Type	Yes	1	n/a	The refund response.
<TransactionId>	String	Yes	1	36	A unique transaction id assigned by the Exactor system.
<TransactionDate>	DateTime	Yes	1	n/a	The date and time that the transaction was processed by the Exactor system.
<RefundDate>	Date	Yes	1	10	The refund date for this committed invoice transaction in YYYY-MM-DD format.
<PriorTransactionId>	String	Yes	1	36	The transaction id that this refund request applies to.

DeleteResponse

Field Name	Data Type	Required	Max #	Width	Description
<DeleteResponse>	DeleteResponse Type	Yes	1	n/a	The delete response.
<TransactionId>	String	Yes	1	36	A unique transaction id assigned by the Exactor system.
<TransactionDate>	DateTime	Yes	1	n/a	The date and time that the transaction was processed by the Exactor system.
<PriorTransactionId>	String	Yes	1	36	The transaction id that this refund request applies to.

ErrorResponse

Field Name	Data Type	Required	Max #	Width	Description
<ErrorResponse>	ErrorResponseType	Yes	1	n/a	The error response.
<LineNumber>	Integer	Yes	1	n/a	The approximate line number of the TaxRequest XML where the error occurred.
<ColumnNumber>	Integer	Yes	1	n/a	The approximate column number of the TaxRequest XML where the error occurred.
<ErrorCode>	String	Yes	1	2	An error code that informs the caller about the type of error that occurred. This will be: 01 : General Error 02 : Missing or Invalid Merchant ID 03 : Missing or Invalid User ID 04 : Missing or Invalid Partner ID 05 : Missing or Invalid Digital Signature 10 : Missing or Invalid Sale Date 11 : Invalid Currency Code 12 : Invalid Exemption ID 13 : Missing or Invalid Ship From Address 14 : Missing or Invalid Ship To Address 15 : Missing Line Items 16 : Invalid SKU code 17 : Missing or Invalid Gross Amount 30 : Missing or Invalid Commit Date 31 : Missing Invoice Number 32 : Missing InvoiceRequest or PriorTransactionId 33 : Transaction Does Not Exist 34 : Transaction Already Committed 40 : Missing or Invalid Refund Date 41 : Missing PriorTransactionId 42 : Transaction Does Not Exist 43 : Transaction Was Never Committed 44 : Transaction Already Refunded 45 : Transaction Already Deleted 46 : Transaction Cannot Be Deleted
<ErrorDescription>	String	Yes	1	1024	A string that informs the caller about the type of error that occurred.

SUPPORTING TYPES

Primitive Types

Field Type	Width	Description	Examples
Date	10	A date YYYY-MM-DD	2006-10-10
DateTime	n/a	A date YYYY-MM-DD T HH:MM:SS	2006-10-10T12:30:00
Decimal	n/a	A number that may include a fraction	0.50, 1, 10.30
Integer	n/a	A whole number	0, 1, 2
String	n/a	A character string	Abc, 123, foo
CurrencyCodeType	3	A currency code	USD, CAD
TaxObligationType	n/a	A tax obligation type	Full, Partial, or None
TaxClassType	n/a	A tax class type	Sales or Use
TaxDirectionType	7	A tax direction type	Forward or Reverse

AddressType

Field Name	Data Type	Required	Max #	Width	Description
FullName	String	Yes	1	255	The full name of the person or company.
Street1	String	Yes	1	255	The first line of the address.
Street2	String	No	1	255	The second line of the address.
City	String	Yes	1	50	The city.
County	String	No	1	50	The county.
StateOrProvince	String	Yes	1	50	The state or province.
PostalCode	String	Yes	1	16	The postal code. This can be a 5 digit zip code, zip + 4 (55555-4444), or a province code.
Country	String	No	1	50	The country. This will default to USA

TaxInfoType

Field Name	Data Type	Required	Max #	Width	Description
AuthorityLevel	AuthorityLevelType	Yes	1	15	The tax authority level. This will be set to one of the following values: <ul style="list-style-type: none"> Country StateOrProvince District County City
AuthorityName	String	Yes	1	255	The tax authority name.

CityOrCountyOr District	String	No	1	255	The city, county, or district where the tax is owed.
StateOrProvince	String	Yes	0	80	The state where the tax is owed.
Country	String	Yes	1	80	The country where the tax is owed.
TaxAmount	Decimal	Yes	1	n/a	The tax amount.
TaxRate	Decimal	Yes	1	n/a	The tax rate.

Examples

Sample Invoice Tax Calculation

The following example shows a simple request to perform a tax calculation. This transaction will be stored in the Exactor system, but will not affect the merchant's tax obligations for reporting, filing, and remittance purposes because it has not been committed to the system. This is useful for displaying what a customer's tax obligations will be before completing the sale. For instance, you may use a tax calculation such as this to display the taxes in the confirmation page of an online shopping cart just before the customer hits the final submit button to confirm the sale.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxRequest xmlns="http://www.exactor.com/ns">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <InvoiceRequest>
    <SaleDate>2010-04-01</SaleDate>
    <CurrencyCode>USD</CurrencyCode>
    <BillTo>
      <FullName>John Smith</FullName>
      <Street1>1600 Market Street</Street1>
      <City>Santa Clara</City>
      <StateOrProvince>CA</StateOrProvince>
      <PostalCode>95050</PostalCode>
      <Country>USA</Country>
    </BillTo>
    <ShipTo>
      <FullName>John Smith</FullName>
      <Street1>1600 Market Street</Street1>
      <City>Santa Clara</City>
      <StateOrProvince>CA</StateOrProvince>
      <PostalCode>95050</PostalCode>
      <Country>USA</Country>
    </ShipTo>
    <ShipFrom>
      <FullName>My Store</FullName>
      <Street1>3600 Hope Street</Street1>
      <City>Los Angeles</City>
      <StateOrProvince>CA</StateOrProvince>
      <PostalCode>90007</PostalCode>
```

```
<Country>USA</Country>
</ShipFrom>
<LineItem id="_1">
  <Description>Books</Description>
  <Quantity>1</Quantity>
  <GrossAmount>55.00</GrossAmount>
</LineItem>
<LineItem id="_2">
  <Description>Clothing</Description>
  <Quantity>1</Quantity>
  <GrossAmount>100.00</GrossAmount>
</LineItem>
</InvoiceRequest>
</TaxRequest>
```

The request above will result in a response being sent back to the caller.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxResponse xmlns="http://www.exactor.com/ns" version="1.0">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <InvoiceResponse>
    <TransactionId>379359fc-4a7e-4586-a107-bcfce7025a43</TransactionId>
    <TransactionDate>2010-04-01T03:09:38.557-08:00</TransactionDate>
    <SaleDate>2010-04-01</SaleDate>
    <CurrencyCode>USD</CurrencyCode>
    <TaxClass>Sales</TaxClass>
    <TaxDirection>Forward</TaxDirection>
    <GrossAmount>155.00</GrossAmount>
    <TotalTaxAmount>12.79</TotalTaxAmount>
    <TaxObligation>Full</TaxObligation>
    <LineItem id="_1">
      <GrossAmount>55.00</GrossAmount>
      <TaxDirection>Forward</TaxDirection>
      <TotalTaxAmount>4.54</TotalTaxAmount>
    </LineItem>
    <LineItem id="_2">
      <GrossAmount>100.00</GrossAmount>
      <TaxDirection>Forward</TaxDirection>
      <TotalTaxAmount>8.25</TotalTaxAmount>
    </LineItem>
  </InvoiceResponse>
</TaxResponse>
```

Sample commit of an invoice that has already been calculated

This example shows how to commit the tax calculation request from the example shown above so that it will be included for all tax reporting, filing, and remittance. If this were an online shopping cart, you would want to make a call such as this to commit the taxes just after the customer hit the confirm button and payment was made. As you can see, we are sending in the transaction ID from the calculation response to commit it to the system along with the commit date and invoice number.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxRequest xmlns="http://www.exactor.com/ns">  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <CommitRequest>
    <CommitDate>2010-04-01</CommitDate>
    <InvoiceNumber>1001</InvoiceNumber>
    <PriorTransactionId>379359fc-4a7e-4586-a107-bcfce7025a43</PriorTransactionId>
  </CommitRequest>
</TaxRequest>
```

The request above will result in a response being sent back to the caller.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxResponse xmlns="http://www.exactor.com/ns" version="1.0">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <CommitResponse>
    <TransactionId>379359fc-4a7e-4586-a107-bcfce7025a43</TransactionId>
    <TransactionDate>2010-04-01T03:09:50.557-08:00</TransactionDate>
    <CommitDate>2010-04-01</CommitDate>
    <InvoiceNumber>1001</InvoiceNumber>
  </CommitResponse>
</TaxResponse>
```

Sample of a combined Invoice and Commit

This example demonstrates how the invoice calculation and commit actions shown in the two examples above can be combined into a single step. This transaction accomplishes the same result as the prior two examples, but only requires one call to the Exactor system. This is useful for cases where the merchant or consumer does not need to review the taxes before committing them to the system and is often used for integrations with financial accounting systems.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxRequest xmlns="http://www.exactor.com/ns">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <CommitRequest>
    <CommitDate>2010-04-01</CommitDate>
    <InvoiceNumber>1001</InvoiceNumber>
    <InvoiceRequest>
      <SaleDate>2010-04-01</SaleDate>
      <CurrencyCode>USD</CurrencyCode>
      <BillTo>
        <FullName>John Smith</FullName>
        <Street1>1600 Market Street</Street1>
        <City>Santa Clara</City>
        <StateOrProvince>CA</StateOrProvince>
        <PostalCode>95050</PostalCode>
        <Country>USA</Country>
      </BillTo>
      <ShipTo>
        <FullName>John Smith</FullName>
        <Street1>1600 Market Street</Street1>
        <City>Santa Clara</City>
        <StateOrProvince>CA</StateOrProvince>
        <PostalCode>95050</PostalCode>
        <Country>USA</Country>
      </ShipTo>
      <ShipFrom>
        <FullName>My Store</FullName>
        <Street1>3600 Hope Street</Street1>
        <City>Los Angeles</City>
      </ShipFrom>
    </InvoiceRequest>
  </CommitRequest>
</TaxRequest>
```



```
<StateOrProvince>CA</StateOrProvince>
<PostalCode>90007</PostalCode>
<Country>USA</Country>
</ShipFrom>
<LineItem id="_1">
  <Description>Books</Description>
  <Quantity>1</Quantity>
  <GrossAmount>55.00</GrossAmount>
</LineItem>
<LineItem id="_2">
  <Description>Clothing</Description>
  <Quantity>1</Quantity>
  <GrossAmount>100.00</GrossAmount>
</LineItem>
</InvoiceRequest>
</CommitRequest>
</TaxRequest>
```

The above request will result in this response.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxResponse xmlns="http://www.exactor.com/ns" version="1.0">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <CommitResponse>
    <TransactionId>f56457f5-349e-4f8d-9a54-df2ae1185c03</TransactionId>
    <TransactionDate>2010-04-01T03:15:38.459-08:00</TransactionDate>
    <CommitDate>2010-04-01</CommitDate>
    <InvoiceNumber>1001</InvoiceNumber>
    <InvoiceResponse>
      <TransactionId>f56457f5-349e-4f8d-9a54-df2ae1185c03</TransactionId>
      <TransactionDate>2010-04-01T03:15:38.459-08:00</TransactionDate>
      <SaleDate>2010-04-01</SaleDate>
      <CurrencyCode>USD</CurrencyCode>
      <TaxClass>Sales</TaxClass>
      <TaxDirection>Forward</TaxDirection>
      <GrossAmount>155.00</GrossAmount>
      <TotalTaxAmount>12.79</TotalTaxAmount>
      <TaxObligation>Full</TaxObligation>
      <LineItem id="_1">
        <GrossAmount>55.00</GrossAmount>
        <TaxDirection>Forward</TaxDirection>
        <TotalTaxAmount>4.54</TotalTaxAmount>
      </LineItem>
      <LineItem id="_2">
        <GrossAmount>100.00</GrossAmount>
        <TaxDirection>Forward</TaxDirection>
        <TotalTaxAmount>8.25</TotalTaxAmount>
      </LineItem>
    </InvoiceResponse>
  </CommitResponse>
</TaxResponse>
```

Sample of a Refund Request

This example shows how to refund a transaction that has already been committed. Refund requests can only be used on transactions that have been committed.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxRequest xmlns="http://www.exactor.com/ns">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <RefundRequest>
    <RefundDate>2010-04-01</RefundDate>
    <PriorTransactionId>379359fc-4a7e-4586-a107-bcfce7025a43</PriorTransactionId>
  </RefundRequest>
</TaxRequest>
```

The request above will result in a response being sent back to the caller.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxResponse xmlns="http://www.exactor.com/ns" version="1.0">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <RefundResponse>
    <TransactionId>379359fc-4a7e-4586-a107-bcfce7025a55</TransactionId>
    <TransactionDate>2010-04-01T03:20:50.557-08:00</TransactionDate>
    <RefundDate>2010-04-01</RefundDate>
    <PriorTransactionId>379359fc-4a7e-4586-a107-bcfce7025a43</PriorTransactionId>
  </RefundResponse>
</TaxResponse>
```

Sample of a Delete Request

This example shows how to delete a transaction. Delete requests can be used on any type of transaction.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxRequest xmlns="http://www.exactor.com/ns">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <DeleteRequest>
    <PriorTransactionId>379359fc-4a7e-4586-a107-bcfce7025a43</PriorTransactionId>
  </DeleteRequest>
</TaxRequest>
```

The request above will result in a response being sent back to the caller.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxResponse xmlns="http://www.exactor.com/ns" version="1.0">
  <MerchantId>99999999</MerchantId>
  <UserId>myusername</UserId>
  <DeleteResponse>
    <TransactionId>379359fc-4a7e-4586-a107-bcfce7025a49</TransactionId>
    <TransactionDate>2010-04-01T03:20:50.557-08:00</TransactionDate>
    <PriorTransactionId>379359fc-4a7e-4586-a107-bcfce7025a43</PriorTransactionId>
  </DeleteResponse>
</TaxResponse>
```

Sample of a Request by a Partner on Behalf of a Merchant

This example shows how an Exactor Partner can submit a transaction on behalf of a merchant. The partner uses its PartnerId and UserId to submit a transaction on behalf of its merchant. All partner requests require a digital signature.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TaxRequest xmlns="http://www.exactor.com/ns">
  <MerchantId>99999999</MerchantId>
  <UserId>partneruser</UserId>
  <PartnerId>88888888</PartnerId>
  <DigitalSignature>gsERd5sdTFFgJzSe38ju135c5ADxgM4CVFPt6m5VhN8=</DigitalSignature>
  <InvoiceRequest>
    <SaleDate>2010-04-01</SaleDate>
    <ShipTo>
      <Street1>123 Easy Street</Street1>
      <Street2>Apt. 1</Street2>
      <City>Los Angeles</City>
      <StateOrProvince>CA</StateOrProvince>
      <PostalCode>90049</PostalCode>
    </ShipTo>
    <ShipFrom>
      <StateOrProvince>WA</StateOrProvince>
      <PostalCode>98007</PostalCode>
    </ShipFrom>
    <LineItem id="_1">
      <GrossAmount>100.00</GrossAmount>
    </LineItem>
  </InvoiceRequest>
</TaxRequest>
```

Sample of a Response by a Partner on Behalf of a Merchant

```
<?xml version="1.0" encoding="UTF-8"?>
<TaxResponse version="1.0" xmlns="http://www.exactor.com/ns">
  <MerchantId>99999999</MerchantId>
  <UserId>partneruser</UserId>
  <PartnerId>88888888</PartnerId>
```

```
<InvoiceResponse>
  <TransactionId>97f28390-9c97-4300-8cb4-1b3b562169d6</TransactionId>
  <TransactionDate>2010-04-01T00:14:23.328-07:00</TransactionDate>
  <SaleDate>2010-04-01</SaleDate>
  <CurrencyCode>USD</CurrencyCode>
  <TaxClass>Sales</TaxClass>
  <TaxDirection>Forward</TaxDirection>
  <GrossAmount>100.00</GrossAmount>
  <TotalTaxAmount>9.75</TotalTaxAmount>
  <TaxObligation>Full</TaxObligation>
  <LineItem id="_1">
    <GrossAmount>100.00</GrossAmount>
    <TaxDirection>Forward</TaxDirection>
    <TotalTaxAmount>9.75</TotalTaxAmount>
  </LineItem>
</InvoiceResponse>
</TaxResponse>
```

SAMPLE CODE FOR GENERATING A DIGITAL SIGNATURE

```
require 'rubygems'
require 'hmac-sha2'

EOL = "\n"

def get_text root, name = nil
  elem = (name ? root.elements[name] : root)

  return " " if elem.nil?
  return " " if elem.text.nil?

  elem.text.tr(EOL, ' ')
end

def walk_address a, req
  a << get_text(req, 'FullName')
  a << get_text(req, 'Street1')
  a << get_text(req, 'Street2')
  a << get_text(req, 'City')
  a << get_text(req, 'County')
  a << get_text(req, 'StateOrProvince')
  a << get_text(req, 'PostalCode')
  a << get_text(req, 'Country')
end

def walk_commit a, req
  a << get_text(req, 'CommitDate')

  unless (inv = req.elements['InvoiceRequest']).nil?
    walk_invoice(a, inv)
  end
end
```

```
    unless (elem = req.elements['PriorTransactionId']).nil?  
      a << get_text(elem)  
    end  
end
```

```
def walk_delete a, req  
  a << get_text(req, 'PriorTransactionId')  
end
```

```
def walk_refund a, req  
  a << get_text(req, 'RefundDate')  
  a << get_text(req, 'PriorTransactionId')  
end
```

```
def walk_invoice a, req  
  a << get_text(req, 'SaleDate')  
  
  walk_address(a, req.elements['ShipTo'])  
  walk_address(a, req.elements['ShipFrom'])  
  
  req.elements.each('LineItem') { |item|  
    a << get_text(item, 'GrossAmount')  
  }  
end
```

```
def walk root  
  a = []  
  
  a << get_text(root, 'MerchantId')  
  a << get_text(root, 'UserId')  
  
  root.elements.each { |req|
```



```

case req.name
  when 'InvoiceRequest'
    walk_invoice(a, req)

  when 'CommitRequest'
    walk_commit(a, req)

  when 'RefundRequest'
    walk_refund(a, req)

  when 'DeleteRequest'
    walk_delete(a, req)
end
}

a.join(EOL) << EOL
end

if __FILE__ == $0
  require 'rexml/document'

  doc = File.open(ARGV.shift) do |rd|
    REXML::Document.new rd
  end

  KEY = 'jylaV8yoWCXCJzS7y8julP5cMANxmM4UBBPt3mnlxN0='.unpack('m*').to_s

  hmac = HMAC::SHA256.new KEY
  hmac << walk(doc.root)

  puts hmac.digest.to_a.pack('m*')
end

```

APPENDIX

CURRENCY CODES

The following table lists countries and currency codes supported in the Exactor System. Merchants must use USD in the United States. Other than that, any currency and country combination may be used.

Country Name	Country Abbreviation	Currency Code	Currency Name
Algeria	DZA	DZD	Dinars
Argentina	ARG	ARS	Pesos
Armenia	ARM	AMD	Drams
Australia	AUS	AUD	Dollars
Austria	AUT	EUR	Euro
Azerbaijan	AZE	AZN	New Manats
Bahrain	BHR	BHD	Dinars
Bangladesh	BGD	BDT	Taka
Belgium	BEL	EUR	Euro
Bolivia	BOL	BOB	Bolivianos
Bosnia and Herzegovina	BIH	BAM	Convertible Marka
Botswana	BWA	BWP	Pulas
Brazil - ICMS	BRA	BRL	Real
Brazil -IPI	BRA	BRL	Real
Brunei	BRN	BND	Dollars
Bulgaria	BGR	BGN	Leva
Canada	CAN	CAD	Dollars
Chile	CHL	CLP	Pesos
China	CHN	CNY	Yuan Renminbi
Colombia	COL	COP	Pesos
Costa Rica	CRI	CRC	Colones
Croatia	HRV	HRK	Kuna
Cyprus	CYP	EUR	Euro
Czech Republic	CZE	CZK	Koruny
Denmark	DNK	DKK	Kroner
Dubai	ARE	AED	Dirhams
Ecuador	ECU	USD	US Dollars
Egypt	EGY	EGP	Pounds
El Salvador	SLV	SVC/USD	Colones/US Dollars

Estonia	EST	EEK	Krooni
Ethiopia	ETH	ETB	Birr
Finland	FIN	EUR	Euro
France	FRA	EUR	Euro
Georgia	GEO	GEL	Lari
Germany	DEU	EUR	Euro
Greece	GRC	EUR	Euro
Guatemala	GTM	GTQ	Quetzales
Hong Kong	HKG	HKD	Dollars
Hungary	HUN	HUF	Forint
India	IND	INR	Rupees
Indonesia	IDN	IDR	Rupiahs
Iraq	IRQ	IQD	Dinars
Ireland	IRL	EUR	Euro
Israel	ISR	ILS	New Shekels
Italy	ITA	EUR	Euro
Japan	JPN	JPY	Yen
Kazakhstan	KAZ	KZT	Tenge
Korea, Republic of	KOR	KPW	Won
Kuwait	KWT	KWD	Dinars
Latvia	LVA	LVL	Lati
Lithuania	LTU	LTL	Litai
Luxembourg	LUX	EUR	Euro
Malaysia	MYS	MYR	Ringgits
Malta	MLT	EUR	Euro
Mexico	MEX	MXN	Pesos
Montenegro	YUG	EUR	Euro
Morocco	MAR	MAD	Dirhams
Netherlands	NLD	EUR	Euro
New Zealand	NZL	NZD	Dollars
Nigeria	NGA	NGN	Nairas
Norway	NOR	NOK	Kroner
Oman	OMN	OMR	Rials
Pakistan	PAK	PKR	Rupees
Panama	PAN	PAB	Balboa
Peru	PER	PEN	Nuevos Soles
Philippines	PHL	PHP	Pesos
Poland	POL	PLN	Zlotych
Portugal	PRT	EUR	Euro
Qatar	QAT	QAR	Rials
Romania	ROM	RON	New Lei

Russia	RUS	RUB	Rubles
Saudi Arabia	SAU	SAR	Riyals
Serbia	YUG	RSD	Dinars
Singapore	SGP	SGD	Dollars
Slovakia	SVK	EUR	Euro
Slovenia	SVN	EUR	Euro
South Africa	ZAF	ZAR	Rand
Spain	ESP	EUR	Euro
Sweden	SWE	SEK	Kronor
Switzerland	CHE	CHF	Francs
Switzerland	CHE	CHF	Francs
Taiwan	TWN	TWD	New Dollars
Thailand	THA	THB	Baht
Tunisia	TUN	TND	Dinars
Turkey	TUR	TRY	Lira
Ukraine	UKR	UAH	Hryvnia
United Kingdom	GBR	GBP	Pounds
United Kingdom	GBR	GBP	Pounds
United States	USA	USD	Dollars
Uruguay	URY	UYU	Pesos
Uzbekistan	UZB	UZS	Sums
Venezuela	VEN	VEF	Bolivares Fuertes
Vietnam	VNM	VND	Dong