# SLIM CD API SUPPORT FOR CONVENIENCE FEES

# Overview

SLIM CD’s payment gateway allows a developer to program convenience fee capability into their services. The convenience fee API requires that rules are created on the SLIM CD website to identify the types of payments that will incur convenience fees, etc.

1. If you wish to support multiple payment acceptance “channels”, such as retail vs mail-order and have different rules per channel, you will need a separate MID/TID for each channel so the rules can be assigned per MID/TID. Note that you can use one set of rules for card-present and a different set for CARD-NOT-PRESENT on a single channel.
2. The convenience fee web service’s rules can be configured to return convenience fees for credit cards, checks, pin debit, gift cards, etc.
3. The convenience fee web service’s rules can be configured such that the fee only applies to a particular brand (VISA/MC, etc) or to a specific type of card by BIN lookup (different rates for BUSINESS cards vs CONSUMER cards, etc)
4. The consumer must be presented with a “confirmation” screen, showing the convenience fee and the new total. If the user does not agree to the new amount, the transaction must be stopped.
5. A convenience fee rule can be configured to allow transactions to continue for payment without a convenience fee. For example, a zero dollar flat fee convenience would NOT charge a convenience fee.
6. Convenience fee rules can be configured to charge a percent of the total or a flat fee.
7. Convenience fee rules can have a “cap” on the fee charged.
8. Convenience fee rules can have a lower value and upper value for the transaction so that “tiers” of fees can be created.

When a convenience fee can be applied, it is up to the developer to replace the amount in the ProcessTransaction with the new/updated amount (that includes the convenience fee) and to send the convenience fee amount in a separate field (see ProcessTransaction documentation for the “amount” and “conveniencefee” fields).

# GetConvenienceFee

We support GetConvenienceFee as follows:

1. In a web service call with NAME/VALUE pairs that returns XML

2. In a Restful web service call with NAME/VALUE pairs or JSON, that returns JSON

3. In the SLIM CD .NET library

4. In the SLIMCD.JS JavaScript library

# GetConvenienceFee Request Fields

The request can contain the following fields:

 username – A SLIM CD login credential or an API ACCESS Credential

 password – Password for a login credential or blank for an API ACCESS Credential

 clientid – The SLIM CD ClientID to use

 siteid - The SLIM CD SiteID to use

 cardnumber – Cardnumber (for keyed transactions)

 trackdata – Encrypted or unencrypted track data (from which to derive the cardnumber)

 emvtags – Encrypted or unencrypted EMV data (from which to derive the cardnumber)

 gateid – A previous transaction from SLIM CD from which to derive the cardnumber

 encryption\_type – Used for specific legacy cases (no longer needed, leave blank)

 paymenttype - The type of payment

* creditcard
* pindebit
* elecrtoniccheck
* papercheck
* giftcard
* ebtcard
* cash
* (empty) means figure out creditcard or debitcard from trackdata, EMVTags and pin indicator

 transtype - The transaction type to be processed (SALE, AUTH, etc)

 amount - The original amount

 cardpresent – Indicator of card present (“yes”,”no”, or “0”,”1”)

 recurring – Indicator of a recurring transaction (“yes”,”no”, or “0”,”1”)

 pin – Indicator of a PIN entry on a transaction (“yes”,”no”, or “0”,”1”)

You MUST send:

 amount – Original amount

paymenttype – must be one of the following:

creditcard , electroniccheck , papercheck , giftcard , pindebit , ebtcard, cash, or an empty string

Note that if you want the slim cd webserivce to decide between creditcard and pindebit, send this field as an empty string. For all other requests, send the appropriate value.

If paymenttype=creditcard, you must send ONE OF:

 cardnumber

 trackdata

 emvtags

 gateid

# URL ENDPOINTS for Web Service Calls

URL to post to for XML:

 url = "https://trans.slimcd.com/soft/getsconveniencefee.asp";

URL to post to for JSON:

 https://trans.slimcd.com/soft/json/jsonscript.asp?service=GetConvenienceFee

# GetConvenienceFee Response Fields

Response contains the standard SLIM CD

 response

 responsecode

 description

 responseurl

 datablock

Inside the datablock are additional fields for:

amount – Updated amount to use in ProcessTransaction and the confirmation screen

 conveniencefee – Calculated convenience fee amount to use in ProcessTransaction

 conveniencefee\_message – Message text to show the user, or reason for the FAIL or ERROR

 conveniencefee\_receiptlabel – Text to show in prompts or on a receipt

(note that if you are including this text in a sentence, you should remove any colon characters before displaying the text in a sentence)

ruleid – a list of comma-delimited rules applied. Note that a single rule will be provide on a successful transaction (indicating the rule that dictated the choice of fee). If multiple rules are identified, an error is returned with a comma-delimited list of ruleid values (showing the multiple rules that matched the request)

# RESPONSES:

If you call the getconveniencefee.asp, you get back XML. Here is an example of the full reply:

### SUCCESS:

<?xml version="1.0" encoding="utf-8"?>

<reply>

 <response>Success</response>

 <responsecode>0</responsecode>

 <description>OK</description>

 <responseurl>/soft/getconveniencefee.asp</responseurl>

 <datablock>

 <amount>103.00</amount>

 <conveniencefee>3.00</conveniencefee>

 < conveniencefee \_message>(3% Convenience Fee Applied)</ conveniencefee \_message>

 <conveniencefee\_receiptlabel>Convenience Fee:</conveniencefee\_receiptlabel>

 <ruleid></ruleid>

</datablock>

</reply>

### FAIL:

<?xml version="1.0" encoding="utf-8"?>

<reply>

 <response>Fail</response>

 <responsecode>1</responsecode>

 <description> No matching record found. Fee not applied </description>

 <responseurl>/soft/get conveniencefee </responseurl>

 <datablock>

 <amount>100.00</amount>

 < conveniencefee >0.00</ conveniencefee >

 < conveniencefee \_message> No matching record found. Fee not applied </ conveniencefee \_message>

 <conveniencefee\_receiptlabel>Convenience Fee:</conveniencefee\_receiptlabel>

 <ruleid></ruleid>

</datablock>

</reply>

### ERROR:

<?xml version="1.0" encoding="utf-8"?>

<reply>

 <response>Error</response>

 <responsecode>2</responsecode>

 <description>Convenience Fee not enabled for this account</description>

 <responseurl>/soft/get conveniencefee.asp</responseurl>

 <datablock>

 <amount>100.00</amount>

 < conveniencefee >0.00</ conveniencefee >

 <s conveniencefee \_message>Convenience Fee not enabled for this account</ conveniencefee \_message>

 <conveniencefee\_receiptlabel>Convenience Fee:</conveniencefee\_receiptlabel>

 <ruleid></ruleid>

</datablock>

</reply>

#### Note that:

 On a SUCCESS RESPONSE, we display the confirmation page to the user

On a FAIL response, we skip informing the user (and the confirmation page) and process the transaction

On an ERROR, we display the error and stop the transaction (the user must turn off convenience fee or contact SLIM CD to configure it on our side)

## JSON:

IF YOU PREFER to receive JSON instead of XML, use the following URL:

https://trans.slimcd.com/soft/json/jsonscript.asp?service=GetConvenienceFee

You can HTTP POST the name/value pairs or the JSON and you will get back JSON.

You will get the same data fields as shown in the XML response.

SUCCESS:

{

                "response": "Success",

                "responsecode": "0",

                "description": "OK",

                "responseurl": "/soft/getconveniencefee.asp",

                "datablock": {

                                "amount": "10.30",

                                "conveniencefee": "0.30",

                                " conveniencefee \_message": "(3% Convenience Fee has been applied)"

                }

}

FAIL:

{

                "response": "Fail",

                "responsecode": "1",

                "description": "(Card not found in CardType/BIN database)",

                "responseurl": "/soft/getconveniencefee.asp",

                "datablock": {

                                "amount": "10.00",

                                " conveniencefee ": "0.00",

                                " conveniencefee \_message": "(Card not found in CardType/BIN database)"

                }

}

ERROR:

{

                "response": "Error",

                "responsecode": "2",

                "description": "Convenience Fee not enabled for this account",

                "responseurl": "/soft/getconveniencefee.asp",

                "datablock": {

                                "amount": "10.00",

                                "conveniencefee": "0.00",

                                "conveniencefee\_message": "Conveneince Fee not enabled for this account"

                }

}

## SLIM CD’s .NET Library

The SLIM CD library has the following

 SlimCD.Transact.GetConvenienceFeeRequest req = new SlimCD.Transact.GetConvenienceFeeRequest() ;

 req.username=”xxx” ;

 req.password=”yyy” ;

 ...

 SlimCD.Transact.GetConvenienceFeeReply reply = SlimCD.Transact.GetConvenienceFee (req) ;

 if (reply.response.ToUpper() == "SUCCESS")

 {

 // Show the message for debugging, this should be replaced with a confirmation screen!

 MessageBox.Show(reply.conveniencefee\_message) ;

 // If agreed, process the transaction. Else stop.

 }

 else if (reply.response.ToUpper() == “FAIL")

 {

 // Just process the transaction without confirmation or convenience fee.

 }

 else

 {

 // Show an error and stop.

 }

## SLIM CD’s JavaScript Library

The SLIM CD JavaScript library has the GetConvenienceFee and follows the same pattern.

Notes:

The cardpresent indicator is used by the rules to determine if cardpresent transactions are permitted to charge a convenience fee. Same for the recurring indicator.