Productive
Computing

FM CreditCard

Functions Guide

Revised November 8, 2013

950 Boardwalk, Suite 205, San Marcos, CA 92078 • (760) 510-1200 • www.productivecomputing.com

Table of Contents

I. Introduction

Description

FM Credit Card is a credit card processing plug-in for FileMaker. With this plug-in you are able to automatically process secure SSL encrypted credit card payments through your gateway all from within FileMaker. The plug-in works with an array of gateways to meet all your needs and provide you with more options. These operations are accomplished using FileMaker function calls from within FileMaker calculations. These calculations are generally determined from within FileMaker "SetField," "SetVariable" or "If" script steps. For a list of the basic integration steps, please see the accompanying Developer's Guide.

Intended Audience

FileMaker developers or persons who have knowledge of FileMaker scripting, calculations and relationships as proper use of the plug-in requires that FileMaker integration scripts be created in your FileMaker solution.

Successful Integration Practices:

1) Read the Developer's Guide
2) Read the Functions Guide
3) Review our FileMaker demo and video tutorials
Demo and video tutorials: http://www.productivecomputing.com/credit-card-processing
4) Familiarize yourself with your credit card gateway

Error Handling:

Any of the plug-in functions may encounter an error during processing. When an error occurs during processing, immediately call the PCCC_GetLastError function in order to obtain a full description of the error or error number. This function returns the error message or error number associated with the last error in order to troubleshoot script or logic failures. Please see PCCC_GetLastError function description in the Functions's Guide and the "Handling Errors" section in the Developer's Guide for further clarification on how to properly trap for errors.

II. Function Descriptions

This section describes the functions that are available with the FM Credit Card plug-in.

1) Error and Registration Related Functions

PCCC_Register(ServerName ; ServerPort ; ServerPage ; LicenseID)

Purpose:
Registers the plug-in with Productive Computing's registration servers. Must call Register function once per user per machine before using the plug-in.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
ServerName	The name of Productive Computing's registration server.	"licensing.productivecomputing.com"	
ServerPort	The port on Productive Computing's registration server to access the registration page.	"80"	
ServerPage	The page that the registration code resides on.	"/PCIReg/pcireg.php"	
LicenseID	The license ID provided with the purchase of a plug-in, otherwise the demo code.	"your license ID" or "DEMO-CC1"	

Return Values:
Returns 0 on success, otherwise an !!ERROR!! or error text. Anything other than a 0 is an error.

Notes & Examples:
Example: PCCC_Register("licensing.productivecomputing.com" ; "80" ; "/PCIReg/pcireg.php" ; "DEMO-CC1")

PCCC_Version(type)

Purpose:
Used to identify the plug-in version and/or plug-in name installed on a machine.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
type	Determines the version string to be returned. The "short" version string includes only the version number of the plug-in. Example: "1.0.1.0". The "long" includes the name of the plug-in and version number. Example: "FM Credit Card 1.0.1.0"	"long" or "short"	short

Return Values:
A short or long version string, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_Version("long")

If parameter is not specified, then the default value returned is short.

PCCC_GetOperatingMode

Purpose:
Returns a string representing the current operating mode of the plug-in.

Dependencies:
None

Parameters:
None

Return Values:
"LIVE", "DEMO", "EXPIRED", "UNREGISTERED", otherwise an !!ERROR!!

Notes & Examples:
none

PCCC_GetLastError(type)

Purpose:
This function retrieves the last error caused by the previously executed function.

Dependencies:
Must have called any other function from the plug-in.

Parameters:

Parameter Name	Purpose	Values	Default Value
type	Determines the format of the error returned.	"text" or "number"	"number"

Return Values:
Error number or text depending on parameter, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_GetLastError("Number")

2) Global Gateway Related Functions

PCCC_SetProcessMode(Mode)

Purpose:
This function sets the process mode of the plug-in. Payment gateways often provide a test account to make sure everything is working before going live with customers. Setting "test" mode will utilize the chosen gateway provider's test url. Setting "live" mode should be used when processing real credit card transactions.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
Mode	To set the operating mode of the gateway provider.	"test" or "live"	"test"

Return Values:
0 on success, otherwise an !!ERROR!!.

Notes & Examples:
PCCC_SetProcessMode("test")

PCCC_GetProcessMode

Purpose:
This function gets the current process mode of the plug-in.

Dependencies:
None

Parameters:
None

Return Values:
test, live, otherwise an !!ERROR!!.

Notes & Examples:
None

PCCC_GetResp(Field)

Purpose:
This function can retrieve the last response from a gateway transaction. It is capable of returning both the raw response from the gateway and a specific parsed out value that corresponds to the provided field parameter.

Dependencies:
A previous call to one of the gateway functions.

Parameters:

Parameter Name	Purpose	Values	Default Value
Field	This is the key for a corresponding value in the gateway's response.	Dependent upon the provider. Please see "Available Keys" at the end of this document.	

Return Values:
The response from the gateway, otherwise an !!ERROR!!

Notes & Examples:
The use of GetResp is highly specific to the utilized gateway provider. Possible key values are provided in the documentation, and can also be directly referenced in the gateway provider's api documentation. Example: PCCC_GetResp("email") would return the customer email from an authorize.net response.

Example: PCCC_GetResp would return the raw response from whatever the gateway provider last called.

3) Authorize.net Related Functions

PCCC_AN_AuthCapt(APILoginID ; TransactionKey ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function does an authorization and capture for funds in one action. There are five minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	This is the login provided by Authorize.net after signing up for their payment gateway.	Provided by Authorize.net	
TransactionKey	This is the transaction key provided by Authorize.net, to verify ownership of the gateway account.	Provided by Authorize.net	
Amount	The desired amount of the transaction.	In the format of "Dollars.Cents"	
CardNum	The Credit card number used in the transaction.	some credit card number (no spaces)	
ExpDate	The expiration date of the supplied credit card.	In the format "MMYYYY"	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
A transaction ID to be used for future reference, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_AN_AuthCapt("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "3.50" ; "6011000000000012" ; "122020")
Example using KeyValues in the OptParam: PCCC_AN_AuthCapt("PCI3u3zM6b" ; "92cFg7PCIG89cY"; "3.50" ; "6011000000000012" ; "122020" ; "x_email=test@test.com" ; "x_address=123 Fake St")

PCCC_AN_Authorize(APILoginID ; TransactionKey ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function does an authorization for the specified amount of funds on a provided credit card. The funds are not captured until the Capture function is called with the returned transID. There are five minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	This is the login provided by Authorize.net, after signing up for their payment gateway.	Provided by Authorize.net	
TransactionKey	This is the transaction key provided by Authorize.net to verify ownership of the gateway account.	Provided by Authorize.net	
Amount	The desired amount of the transaction.	In the format of "Dollars.Cents"	
CardNum	The credit card number used in the transaction.	some credit card number (no spaces)	
ExpDate	The expiration date of the supplied credit card.	In the format "MMYYYY"	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
A transaction ID upon a successful authorization, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_AN_Authorize("PCI3u3zM6b" ; "92cFg7PCIG89cY"; "3.50" ; "6011000000000012" ; "122020")

Example using KeyValues in the OptParam: PCCC_AN_Authorize("PCI3u3zM6b" ; "92cFg7PCIG89cY"; "3.50" ; "6011000000000012" ; "122020" ; "x_email=test@test.com" ; "x_address=123 Fake St")

PCCC_AN_Capture(APILoginID ; TransactionKey ; TransID ; Amount; OptParam)

Purpose:
This function can capture funds from a previous call to Authorize. Simply supply the previous transaction ID and an amount less than or equal to the authorized amount. There are four minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	This is the login provided by Authorize.net, after signing up for their payment gateway.	Provided by Authorize.net	
TransactionKey	This is the transaction key provided by Authorize.net, to verify ownership of the gateway account.	Provided by Authorize.net	
TransID	This is the transaction ID from a previous call to PCCC_AN_Capture.	some transaction ID	
Amount	An amount less than or equal to the previously authorized amount.	In the format of "Dollars.Cents"	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
The provided transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_AN_Capture("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "123456" ; "20.00")

Example using KeyValues in the OptParam: PCCC_AN_AuthCapt("PCI3u3zM6b" ; "92cFg7PCIG89cY"; "123456" ; "3.50" ; "x_email=test@test.com" ; "x_ship_to_address=123 Fake St")

PCCC_AN_CaptOnly(APILoginID ; TransactionKey ; AuthCode ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function can capture payment on a transaction that was not authorized through the gateway, but was authorized directly by the bank. In this case, the bank or authorizing party must supply an authorization code to be providing in the function parameter. There are six minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
A previously authorized transaction, by means other than the gateway.

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	This is the login provided by Authorize.net, after signing up for their payment gateway.	Provided by Authorize.net	
TransactionKey	This is the transaction key provided by Authorize.net, to verify ownership of the gateway account.	Provided by Authorize.net	
AuthCode	A code provided by a bank for a previously authorized amount.	some Auth code	
Amount	An amount less than or equal to the previously authorized amount.	In the format of "Dollars.Cents"	
CardNum	The Credit card number used in the transaction.	some credit card number (no spaces)	
ExpDate	The expiration date of the supplied credit card.	In the format "MMYYYY"	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_AN_CaptOnly("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "123456" ; "10.00" ; "6011000000000012" ; "122020")

Example using KeyValues in the OptParam: PCCC_AN_CaptOnly("PCI3u3zM6b" ; "92cFg7PCIG89cY"; "123456" ; "3.50" ; "6011000000000012" ; "122020" ; "x_email=test@test.com" ; "x_address=123 Fake St")

PCCC_AN_Credit(APILoginID ; TransactionKey ; TransID ; Amount ; CardNum; OptParam)

Purpose:
This function can credit or refund a customer for a previous transaction performed on the gateway. There are five minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
Must have a transaction ID from a previous call to AuthCapt or Capture, and that payment must have cleared.

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	This is the login provided by Authorize.net, after signing up for their payment gateway.	Provided by Authorize.net	
TransactionKey	This is the transaction key provided by Authorize.net, to verify ownership of the gateway account.	Provided by Authorize.net	
TransID	This is the transaction ID from a previous call to PCCC_AN_Capture.	some transaction ID	
Amount	An amount less than or equal to the previously authorized amount.	some amount	
CardNum	The Credit card number used in the transaction.	Either the entire credit card number, or just the last four digits.	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_AN_Credit("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "123456" ; "15.00" ; "6011000000000012")

Example using KeyValues in the OptParam: PCCC_AN_Credit("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "123456" ; "15.00" ; "6011000000000012" ; "x_email=test@test.com" ; "x_address=123 Fake St")

PCCC_AN_UCredit(APILoginID ; TransactionKey ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function can refund any credit card without a previous transaction ID. The "U" stands for "Unlinked". Meaning it does not require a previous transaction through the gateway, and it requires special privileges from the merchant's bank to use. Also that is why it requires extra parameters like the full card number, amount and expiration date. There are five minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	This is the login provided by Authorize.net, after signing up for their payment gateway.	Provided by Authorize.net	
TransactionKey	This is the transaction key provided by Authorize.net, to verify ownership of the gateway account.	Provided by Authorize.net	
Amount	The amount to be refunded to a customer's card.	In the format of "Dollars.Cents"	
CardNum	The Credit card number used in the transaction.	some credit card number (no spaces)	
ExpDate	The expiration date of the supplied credit card.	In the format "MMYYYY"	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_AN_UCredit("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "1.00" ; "6011000000000012" ; "122020")

Example using KeyValues in the OptParam: PCCC_AN_UCredit("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "1.00" ; "6011000000000012" ; "122020" ; "x_email=test@test.com" ; "x_address=123 Fake St")

PCCC_AN_Void(APILoginID ; TransactionKey ; TransID; OptParam)

Purpose:
This function can be used to void a previous transaction that has not yet been cleared by the bank, which usually happens at the end of the day. It can not void partial amounts, only the entire amount of the transaction. There are three minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
A previous call to AuthCapt, Authorize, or Capture, as long as it is before the payment is cleared by the bank.

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	This is the login provided by Authorize.net, after signing up for their payment gateway.	Provided by Authorize.net	
TransactionKey	This is the transaction key provided by Authorize.net, to verify ownership of the gateway account.	Provided by Authorize.net	
TransID	This is the transaction ID from a previous call to PCCC_AN_AuthCapt, PCCC_AN_Authorize, or PCCC_AN_Capture.	some transaction key	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
It is always a good idea to call this function before calling PCCC_AN_Credit, if there is any uncertainty about whether the bank has cleared the payment.

Example: PCCC_AN_Void("PCI3u3zM6b" ; "92cFg7PCIG89cY"; "someTransID")

Example using KeyValues in the OptParam: PCCC_AN_Void("PCI3u3zM6b" ; "92cFg7PCIG89cY"; "someTransID" ; "x_email=test@test.com" ; "x_address=123 Fake St")

PCCC_AN_RawPost(APILoginID ; TransactionKey; OptParam)

Purpose:
This is an advanced function that allows for a completely customized post request. There are two minimum parameters and one optional parameter. The OptParam can append any of the key value pairs from the Advanced Integration Method (AIM) developer guide provided by Authorize.net.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
APILoginID	The authorize merchant ID	Provided by Authorize.net	
TransactionKey	The transaction key provided by authorize to validate to gateway posts.	Provided by Authorize.net	
OptParam	Append key value pairs	Please see "AIM_guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
PCCC_AN_RawPost("PCI3u3zM6b" ; "92cFg7PCIG89cY" ; "x_method=ECHECK" ; "x_bank_aba_code=123456789" ; "x_bank_acct_num=8675309" ; "x_bank_acct_type=CHECKING" ; "x_bank_name=chase" ; "x_bank_acct_name=mine" ; "x_echeck_type=ARC" ; "x_bank_check_number=1234")

This is an example of implementing an electronic check transaction that cannot be implemented with the standard functions.

4) PayPal Related Functions

PCCC_PP_Sale(Partner ; Vendor ; User ; Password ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function does an authorization and capture for funds in one action. Besides its seven minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
Amount	The amount of the transaction.	The format should be "dollars.cents", with no commas or dollar signs.	
CardNum	The customer's credit card number.	The number should have no dashes "-", only numbers.	
ExpDate	The expiration date on the card.	The date should be formatted as MMYYYY.	
OptParam	Append key value pairs.	Please see "PP_PayflowPro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_PP_Sale("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "10.01" ; "5105105105105100" ; "1220")

Example using KeyValues in the OptParam: PCCC_PP_Sale("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "10.01" ; "5105105105105100" ; "1220" ; "STREET=123 Fake St." ; "EMAIL=test@test.com")

PCCC_PP_Authorize(Partner ; Vendor ; User ; Password ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function does an authorization for the specified amount of funds on a provided credit card. The funds are not captured until the Capture function is called with the returned origID. Besides its seven minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
Amount	The amount of the transaction.	The format should be "dollars.cents", with no commas or dollar signs.	
CardNum	The customer's credit card number.	The number should have no dashes "-", only numbers.	
ExpDate	The expiration date on the card.	The date should be formatted as MMYYYY.	
OptParam	Append key value pairs	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_PP_Authorize("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "10.01" ; "5105105105105100" ; "1220")

Example using KeyValues in the OptParam: PCCC_PP_Authorize("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "10.01" ; "5105105105105100" ; "1220" ; "STREET=123 Fake St." ; "EMAIL=test@test.com")

PCCC_PP_Capture(Partner ; Vendor ; User ; Password ; OrigID; OptParam)

Purpose:
This function can capture funds from a previous call to Authorize. Simply supply the original ID. Besides its five minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
Must have called PCCC_PP_Authorize and retained the ID returned.

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
OrigID	The ID returned from an original PCCC_PP_Authorize transaction.	some ID	
OptParam	Append key value pairs.	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
A particularly important key value pair for this function is "AMT". This represents the amount. If less than the initial authorized amount is to be captured this is the parameter used to do it, as shown in the example.

Example: PCCC_PP_Capture("PayPal" ; "ProductiveComputing" ; "someUser2" ; "somePassword" ; "123")Or with an amount less than the originally authorized amount:PCCC_PP_Capture("PayPal" ; "ProductiveComputing" ; "ProductiveComputing" ; "PcProgrammer1" ; "123")

Example using KeyValues in the OptParam: PCCC_PP_Capture("PayPal" ; "ProductiveComputing" ; "someUser2" ; "somePassword" ; "123")Or with an amount less than the originally authorized amount:PCCC_PP_Capture("PayPal" ; "ProductiveComputing" ; "ProductiveComputing" ; "PcProgrammer1" ; "123" ; "AMT=10.50")

PCCC_PP_Void(Partner ; Vendor ; User ; Password ; OrigID; OptParam)

Purpose:
This function can void a previous call to Authorize. Simply supply the original ID. Besides its five minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
Must have called PCCC_PP_Authorize and be able to provide the original ID from that transaction.

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
OrigID	The ID returned from an original PCCC_PP_Authorize transaction.	some ID	
OptParam	Append key value pairs.	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_PP_Void("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "123456")

Example using KeyValues in the OptParam: PCCC_PP_Void("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "123456" ; "STREET=123 Fake St." ; "EMAIL=test@test.com")

PCCC_PP_Credit(Partner ; Vendor ; User ; Password ; OrigID; OptParam)

Purpose:
This function can credit or refund a customer for a previous transaction performed on the gateway. Besides its five minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
A previous successful call to either PCCC_PP_Sale or PCCC_PP_Capture.

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
OrigID	The ID from a previously captured payment.	some ID	
OptParam	Append key value pairs.	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_PP_Credit("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "123456")

Example using KeyValues in the OptParam: PCCC_PP_Credit("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "123456" ; "STREET=123 Fake St." ; "EMAIL=test@test.com")

PCCC_PP_UCredit(Partner ; Vendor ; User ; Password ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function can credit or refund a customer for a previous transaction NOT performed on the gateway. The 'U' stands for unlinked, meaning no previous transaction is required to add funds to a card. This is why all of the card information is required. Also, to utilize this function requires special permission from the merchant's bank. Besides its seven minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
Amount	The amount to be refunded to a customer's credit card.	The format should be "dollars.cents", with no commas or dollar signs.	
CardNum	The customer's card number.	The number should have no dashes "-", only numbers.	
ExpDate	The expiration date on the customer's card.	The date should be formatted as MMYYYY.	
OptParam	Append key value pairs.	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_PP_UCredit("PayPal" ; "ProductiveComputing" ; "somUser1" ; "somePassword" ; "10.00" ; "5105105105105100" ; "122020")

Example using KeyValues in the OptParam: PCCC_PP_UCredit("PayPal" ; "ProductiveComputing" ; "somUser1" ; "somePassword" ; "10.00" ; "5105105105105100" ; "122020" ; "STREET=123 Fake St." ; "EMAIL=test@test.com")

PCCC_PP_VoiceAuth(Partner ; Vendor ; User ; Password ; AuthCode ; Amount ; CardNum ; ExpDate; OptParam)

Purpose:
This function can capture payment for a transaction that was voice authorized with the customer's bank. The bank provides an authorization code that is then supplied as one of the parameters to the function. Besides its eight minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
AuthCode	An authorization code provided over the phone by the customer's bank.	some authorization code	
Amount	The amount to be captured from the customer's card.	The format should be "dollars.cents", with no commas or dollar signs.	
CardNum	The customer's card number.	The number should have no dashes "-", only numbers.	
ExpDate	The expiration date on the customer's card.	The date should be formatted as MMYYYY.	
OptParam	Append key value pairs.	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_PP_VoiceAuth("PayPal" ; "ProductiveComputing" ; "somUser1" ; "somePassword" ; "123456" ; "10.00" ; "5105105105105100" ; "122020")

Example using KeyValues in the OptParam: PCCC_PP_VoiceAuth("PayPal" ; "ProductiveComputing" ; "somUser1" ; "somePassword" ; "123456" ; "10.00" ; "5105105105105100" ; "122020" ; "STREET=123 Fake St." ; "EMAIL=test@test.com")

PCCC_PP_Inquire(Partner ; Vendor ; User ; Password ; OrigID; OptParam)

Purpose:
This function can inquire as to the status of a previous transaction. Simply provide it with the ID from any previous call to a PayPal function. Besides its five minimum parameters, it can append any of the key value pairs from the PP_PayflowPro_Guide provided by PayPal.

Dependencies:
Any previous call to a paypal function that returned a transaction ID.

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
OrigID	The ID returned from a previous transaction.	some ID	
OptParam	Append key value pairs.	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_PP_Inquire("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "123456")

Example using KeyValues in the OptParam: PCCC_PP_Inquire("PayPal" ; "ProductiveComputing" ; "someUser1" ; "somePassword" ; "123456"; "STARTTIME=20081220202020" ; "ENDTIME=20091220202020")

PCCC_PP_RawPost(Partner ; Vendor ; User ; Password; OptParam)

Purpose:
This is an advanced function that allows for a custom post to be built. It only requires the login information for a merchant's account, and the rest must be constructed from key value pairs as outlined in the PP_PayflowPro_Guide provided by PayPal. Any of the other built in functions can be recreated with this function as well as being able to use some advanced capabilities not achievable with the built in functions.

Dependencies:
Depends on the transaction built.

Parameters:

Parameter Name	Purpose	Values	Default Value
Partner	PayflowPro login info.	Provided by PayPal	
Vendor	PayflowPro login info.	Provided by PayPal	
User	PayflowPro login info.	Provided by PayPal	
Password	PayflowPro login info.	Provided by PayPal	
OptParam	Append key value pairs.	Please see "PP_Payflowpro_Guide"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
PCCC_PP_RawPost("PayPal"; "ProductiveComputing" ; "someUser2" ; "somePassword" ; "AMT=100.00" ; "TRXTYPE=S" ; "TENDER=D" ; "ACCT=4444333322221111" ; "EXPDATE=1012")

This is an example of a pinless debit transaction. Notice the "TENDER" tag, which represents the method of payment for PayPal.

5) Ogone Related Functions

PCCC_OG_Sale(PSPID ; USERID ; PSWD ; OrderID ; Amount ; Currency ; CardNum ; ExpDate ; CVC; OptParam)

Purpose:
This function provides authorization and capture of the customer's funds in one action. Besides its nine minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
OrderID	A merchant defined order ID.	Must define a unique id for each order.	
Amount	The amount of the transaction.	The amount must either be in the format "dollars.cents" or an integer representing the lowest value of the currency. For example: 1 dollar = "100"	
Currency	The currency of the transaction.	The currentcy as specified by the three character country code. For example: "USD" or "EUR"	
CardNum	The customer's credit card number.	The card number must have no dashes "-", only numeric characters	
ExpDate	The expiration date of the customer's card.	The expiration date on the customer's card in the format MMYYYY	
CVC	The 3 or 4 digit security code on the back of the customer's card.	some security code	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_Sale("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456789" ; "10.00" ; "EUR" ; "4111111111111111" ; "122020" ; "111")

Example using KeyValues in the OptParam: PCCC_OG_Sale("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456789" ; "10.00" ; "EUR" ; "4111111111111111" ; "122020" ; "111" ; "Owneraddress=123 Fake St." ; "EMAIL=test@test.com")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_Authorize(PSPID ; USERID ; PSWD ; OrderID ; Amount ; Currency ; CardNum ; ExpDate ; CVC; OptParam)

Purpose:
This function provides authorization and an ID for later capture of a customer's funds. Besides its nine minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
OrderID	A merchant defined order ID.	Must define a unique id for each order.	
Amount	The amount of the transaction.	The amount must either be in the format "dollars.cents" or an integer representing the lowest value of the currency. For example: 1 dollar = "100"	
Currency	The currency of the transaction. Refer to the ogone currency sheet for options.	The three character currency codes. For example: "USD" or "EUR"	
CardNum	The customer's credit card number.	The card number must have no dashes "-", only numeric characters	
ExpDate	The expiration date on the customer's card.	The expiration date on the customer's card in the format MMYYYY	
CVC	The 3 to 4 digit security code on the back of the card.	some security code	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_Authorize("ProductiveComputing" ; "PCPay" ; "PCI123"; "123456789" ; "10.00" ; "EUR" ; "4111111111111111" ; "122020" ; "111")

Example using KeyValues in the OptParam: PCCC_OG_Authorize("ProductiveComputing" ; "PCPay" ; "PCI123"; "123456789" ; "10.00" ; "EUR" ; "4111111111111111" ; "122020" ; "111" ; "Owneraddress=123 Fake St." ; "EMAIL=test@test.com")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_Capture(PSPID ; USERID ; PSWD ; PAYID ; Amount ; Final; OptParam)

Purpose:
This function is used to capture funds from a previous PCCC_OG_Authorize call. The amount can be less than or equal to the original authorized amount. The final parameter determines whether transaction will be closed for further processing. If false, further transactions can be made with that ID. For example, an initial authorized amount of forty dollars could be captured twice with amounts of twenty dollars. Once finalized, funds can no longer be captured from a given transaction. Besides its six minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
Must have made a previous call PCCC_OG_Authorize and retained the original ID number.

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
PAYID	The ID from the original transaction.	It is also possible to leave PAYID blank "" and provide the OrderID defined by the merchant in a key value pair. However, Ogone recommends the use of the PAYID.	
Amount	An amount less than or equal to the initial authorized amount.	The amount must either be in the format "dollars.cents" or an integer representing the lowest value of the currency. For example: 1 dollar = "100"	
Final	Whether or not further transactions can be made corresponding to the initial transaction ID.	"true" or "false". True meaning the transaction is final, false meaning further transactions can be made corresponding to the given ID.	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_Capture("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456" ; "3.50" ; "true")

Example using KeyValues in the OptParam: PCCC_OG_Capture("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456" ; "3.50" ; "true" ; "Owneraddress=123 Fake St." ; "EMAIL=test@test.com")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_DelAuth(PSPID ; USERID ; PSWD ; PAYID ; Amount ; Final; OptParam)

Purpose:
This function deletes the corresponding authorization. The final parameter determines whether or not the authorization can be renewed with a call to PCCC_OG_Renew. A value of false would allow for later renewal of the authorization, possibly with a different amount. The amount parameter is not explicitly required, but recommended for validation of the original amount. Besides its six minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
Must have a made a previous call to PCCC_OG_Authorize and retained the original ID.

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
PAYID	The ID from the original transaction.	It is also possible to leave PAYID blank "" and provide the OrderID defined by the merchant in a key value pair. However, Ogone recommends the use of the PAYID.	
Amount	The amount of the original authorization.	The amount must either be in the format "dollars.cents" or an integer representing the lowest value of the currency. For example: 1 dollar = "100"	
Final	Whether or not the deletion of the authorization is final.	"true" or "false"	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_DelAuth(""ProductiveComputing" ; "PCPay" ; "PCI123"; "123456" ; "3.50" ; "true")

Example using KeyValues in the OptParam: PCCC_OG_DelAuth(""ProductiveComputing" ; "PCPay" ; "PCI123"; "123456" ; "3.50" ; "true" ; "Owneraddress=123 Fake St." ; "EMAIL=test@test.com")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_Refund(PSPID ; USERID ; PSWD ; PAYID ; Amount ; Final; OptParam)

Purpose:
This function refunds the customer from a previous transaction. The final parameter determines whether further funds can be refunded to that transaction. For example, a transaction that included two items might be refunded for the first item one week, and then later need to be refunded for the second. Setting the final parameter to "false" would allow such a case to be possible. Besides its six minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
A previous call to Sale or Capture and the original ID.

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
PAYID	The ID from the original transaction.	It is also possible to leave PAYID blank "" and provide the OrderID defined by the merchant in a key value pair. However, Ogone recommends the use of the PAYID.	
Amount	The amount of the original sale.	The amount must either be in the format "dollars.cents" or an integer representing the lowest value of the currency. For example: 1 dollar = "100"	
Final	Whether or not the refund is the final transaction for the corresponding ID.	"true" or "false"	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_Refund(""ProductiveComputing" ; "PCPay" ; "PCI123"; "123456" ; "3.50" ; "true")

Example using KeyValues in the OptParam: PCCC_OG_Refund(""ProductiveComputing" ; "PCPay" ; "PCI123"; "123456" ; "3.50" ; "true" ; "Owneraddress=123 Fake St." ; "EMAIL=test@test.com")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_Renew(PSPID ; USERID ; PSWD ; PAYID ; Amount; OptParam)

Purpose:
This function can renew a previously deleted or expired authorization. Besides its five minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
Needs a PAYID from a previously deleted or expired Authorization.

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
PAYID	The ID from the original transaction.	It is also possible to leave PAYID blank "" and provide the OrderID defined by the merchant in a key value pair. However, Ogone recommends the use of the PAYID.	
Amount	The amount of the original authorization.	The amount must either be in the format "dollars.cents" or an integer representing the lowest value of the currency. For example: 1 dollar = "100"	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_Renew(""ProductiveComputing" ; "PCPay" ; "PCI123"; "123456" ; "3.50")

Example using KeyValues in the OptParam: PCCC_OG_Renew(""ProductiveComputing" ; "PCPay" ; "PCI123"; "123456" ; "3.50" ; "Owneraddress=123 Fake St." ; "EMAIL=test@test.com")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_URefund(PSPID ; USERID ; PSWD ; OrderID ; Amount ; Currency ; CardNum ; ExpDate ; CVC; OptParam)

Purpose:
This function can refund any credit card for transactions performed outside of the gateway. It requires special permission from the merchant's bank. The "U" stands for unlinked, meaning the refund is not linked to a previous gateway transaction. Besides its nine minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
OrderID	A merchant defined order id.	Must define a unique id for each order.	
Amount	The amount to be refunded to the customer's card.	The amount must either be in the format "dollars.cents" or an integer representing the lowest value of the currency. For example: 1 dollar = "100"	
Currency	The currency of the transaction. Refer to the ogone currency sheet for options.	The three character currency codes. For example: "USD" or "EUR"	
CardNum	The customer's credit card number.	The card number must have no dashes "-", only numeric characters	
ExpDate	The expiration date on the customer's card.	The expiration date on the customer's card in the format MMYYYY	
CVC	The 3 or 4 digit security code on the back of the customer's card.	some security code	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_URefund("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456" ; "3.50" ; "EUR" ; "4444333322221111" ; "122020" ; "123")

Example using KeyValues in the OptParam: PCCC_OG_URefund("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456" ; "3.50" ; "EUR" ; "4444333322221111" ; "122020" ; "123" ; "Owneraddress=123 Fake St." ; "EMAIL=test@test.com")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_Query(PSPID ; USERID ; PSWD ; PAYID; OptParam)

Purpose:
This function can query the status of any previous transaction that returned a PAYID. Besides its four minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
Any previous call to an Ogone gateway function and the corresponding original ID.

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
PAYID	The ID of the original transaction.	It is also possible to leave PAYID blank "" and provide the OrderID defined by the merchant in a key value pair. However, Ogone recommends the use of the PAYID.	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_OG_Query("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456")

Example using KeyValues in the OptParam: PCCC_OG_Query("ProductiveComputing" ; "PCPay" ; "PCI123" ; "123456" ; "orderID=654321")

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

PCCC_OG_RawPost(PSPID ; USERID ; PSWD ; Type; OptParam)

Purpose:
This is an advanced function that allows for the creation of a custom post, utilizing any of the key value pairs defined in the Ogone direct link guide. Any of the other Ogone built in functions can be recreated with this function, along with added advanced capabilities not achievable with the build in functions. This function only requires the Ogone login parameters, and the rest must be defined by the developer. The type parameter is required because Ogone uses multiple urls to post data depending on the type of transaction. Besides its four minimum parameters, it can append any of the key value pairs from the Ogone_DirectLink guide provided by Ogone.

Dependencies:
Depends on the type of operation created by the developer.

Parameters:

Parameter Name	Purpose	Values	Default Value
PSPID	Ogone login information.	Provided by Ogone	
USERID	Ogone login information.	Provided by Ogone	
PSWD	Ogone login information.	Provided by Ogone	
Type	This represents the three possible transaction types provided by the Ogone gateway. Each transaction type has multiple specific operations to choose from as outlined in the Ogone direct link guide.	"new", "main" or "query"	
OptParam	Append key value pairs.	Please see "Ogone_DirectLink_EN"	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
PCCC_OG_RawPost("ProductiveComputing" ; "PCPay" ; "PCI123" ; "new" ; "amount=120" ; "CARDNO=4444333322221111")

This is an example of implementing a xxx that can not be implemented with the standard functions.

Please note Ogone has been deprecated and is no longer supported in plug-in versions 1.0.2.1 (Windows), 1.0.2.6 (Mac) and above.

6) Eway Related Functions

PCCC_EW_Sale(CustomerID ; TotalAmount ; CardHoldersName ; CardNumber ; ExpMonth ; ExpYear)

Purpose:
This function provides the minimum parameters to process a sale on the Eway payment gateway.

Dependencies:
None

Parameters:

Parameter Name	Purpose	Values	Default Value
CustomerID	Eway merchant ID.	Provided by Eway	
TotalAmount	The amount of the transaction.	some amount	
CardHoldersName	The customer's name as it appears on the card.	some customer name	
CardNumber	The number on the customer's credit card.	some credit card number	
ExpMonth	The expiration month.	in the format MM	
ExpYear	The expiration year.	in the format YY	

Return Values:
A transaction ID on success, otherwise an !!ERROR!!

Notes & Examples:
Example: PCCC_EW_Sale("87654321" ; "10.00" ; "John Doe" ; "4444333322221111" ; "12" ; "20")

PCCC_EW_Refund(CustomerID ; TotalAmount ; ExpMonth ; ExpYear ; OrigTrxnNum ; RefundPassword)

Purpose:
This function can refund a customer from a previous transaction.

Dependencies:

Parameters:

Parameter Name	Purpose	Values	Default Value
CustomerID	Eway merchant ID.	Provided by Eway	
TotalAmount	The amount to be refunded to the customer.	some amount	
ExpMonth	The expiration month on the customer's card.	in the format MM	
ExpYear	The expiration year on the customer's card.	in the format YY	
OrigTrxnNum	The transaction number returned from the original call to PCCC_EW_Sale	some transaction number	
RefundPassword	The password defined by the merchant on the gateway.	some password	

Return Values:

Notes & Examples:
Example: PCCC_EW_Refund("87654321" ; "10.00" ; "12" ; "20" ; "123456" ; "654321")

III. Available Keys for PCCC_GetResp

PCCC_GetResp can be called with no parameters, which will return the raw response from the gateway. PCCC_GetResp can also be called with one of the following key values relative to the gateway provider. The set of key values used corresponds directly to the last gateway call made. If a call was made to Authorize.net, then one of the Authorize.net keys should be used. If a wrong key is ever used, then !!ERROR!! will be returned. The Authorize.net keys are NOT case sensitive, while the PayPal, Ogone and Eway keys ARE CASE SENSITIVE. All available keys are listed below. If there is a missing key that you would like to access, then please email a request to support@productivecomputing.com

Authorize.net Keys

The "API Return Field" is provided purely for reference. Only the Key should be used with the PCCC_GetResp function. Authorize responds with a comma separated list instead of a key value list. The "API Return Field" then allows the developer to identify which key corresponds to a given value in the list, when examining the raw return string.

Key	API Return Field
RespCode	1
RespSubCode	2
RespReasonCode	3
RespReasonText	4
AuthCode	5
AVSResp	6
TransID	7
InvoiceNum	8
Description	9
Amount	10
Method	11
TransType	12
CustID	13
FirstName	14
LastName	15
Company	16
Address	17
City	18
State	19
Zip	20
Country	21
Phone	22
Fax	23
Email	24
ShipToFirsName	25
ShipToLastName	26
ShipToCompany	27
ShipToAddress	28
ShipToCity	29
ShipToState	30
ShipToZip	31
ShipToCountry	32
Tax	33
Duty	34
Freight	35
TaxExempt	36
PONum	37
MD5Hash	38
CCVResp	39
CAVVResp	40
AccountNum	51
CardType	52
SplitTendID	53
RequestedAmount	54
BalanceOnCard	55

PayPal Keys

Possible transaction return values are as follows:

PNREF – Payflow Transaction ID, a unique number that identifies the transaction. Character length and limitations: 12 alphanumeric characters

PPREF – Unique transaction ID of the payment. Character length and limitations: 17-character string

RESULT – The outcome of the attempted transaction. RESULT=0 means the transaction was approved. NOTE: For Account Verification transactions, RESULT=0 with RESPMSG=Verified means a zero dollar authorization has been successfully performed. Any other value for RESULT indicates a decline or error. Character length and limitations: variable length, numeric

CVV2MATCH – Result of the card security code (CVV2) check. The issuing bank may decline the transaction if there is a mismatch. In other cases, the transaction may be approved despite a mismatch. Character length and limitations: 1 alpha character (Y, N, X, or no response)

RESPMSG – The response message returned with the transaction result. Exact wording varies. Sometimes a colon appears after the initial RESPMSG followed by more detailed information. NOTE: For Account Verification transactions, RESULT=0 with RESPMSG=Verified means a zero dollar authorization has been successfully performed. Character length and limitations: variable, alphanumeric characters

AUTHCODE – Returned for Sale, Authorization, and Voice Authorization credit card transactions. AUTHCODE is the approval code obtained over the telephone from the processing network. AUTHCODE is required when submitting a Force (F) transaction. Character length and limitations: 6 alphanumeric characters

AVSADDR – Address Verification Service address response returned if you are using Address Verification Service. Address Verification Service address responses are for advice only. This process does not affect the outcome of the authorization. Character length and limitations: 1 alpha character (Y, N, X, or no response)

AVSZIP – Address Verification Service address response returned if you are using Address Verification Service. Address Verification Service address responses are for advice only. This process does not affect the outcome of the authorization. Character length and limitations: 1 alpha character (Y, N, X, or no response)

IAVS – International Address Verification Service address responses may be returned if you are using Address Verification Service. IAVS responses are for advice only. This value does not affect the outcome of the transaction. Indicates whether Address Verification Service response is international (Y), US (N), or cannot be determined (X). Client version 3.06 or later is required. Character length and limitations: 1 alpha character (Y, N, X, or no response)

PROCAVS – Address Verification Service response from the processor when you use Address Verification Service and send a VERBOSITY request parameter value of MEDIUM. Character length and limitations: 1 character

PROCCVV2 – CVV2 response from the processor when you send a VERBOSITY request parameter value of MEDIUM. Character length and limitations: 1 character

CORRELATIONID – Value used for tracking this Direct Payment transaction.
Character length and limitations: 13 alphanumeric characters

AMEXID – Unique transaction ID returned when VERBOSITY=MEDIUM or
VERBOSITY=HIGH for tracking American Express CAPN transactions. NOTE: American Express CAPN transactions only: used by merchants who authorize transactions through the payflow gateway but settle through a third-party solution. Character length and limitations: 15 numeric characters

AMEXPOSTDATA – Value returned when VERBOSITY=MEDIUM or VERBOSITY=HIGH.
Character length and limitations: 12 alphanumeic characters

Ogone Keys

Possible Order Return values are:

orderID	Your order reference.
PAYID	Payment reference in Ogone's system.
NCSTATUS	First digit of NCERROR.
NCERROR	Error Code.
NCERRORPLUS	Explanation of the error code.
ACCEPTANCE	Acceptance code returned by acquirer.
STATUS	Transaction status.
ECI	Electronic Commerce Indicator.
amount	Order amount (not multiplied by 100).
currency	Order currency.
PM	Payment method.
BRAND	Card brand or similar information for other payment methods.

Possible Maint Return values are:

orderID	Your order reference.
PAYID	Payment reference in our system.
PAYIDSUB	The history level ID of the maintenance operation on the PAYID.
ACCEPTANCE	Acceptance code returned by acquirer.
STATUS	Transaction status.
NCERROR	Error code.
NCSTATUS	First digit of NCERROR.
NCERRORPLUS	Explanation of the error code.
amount	Order amount (not multiplied by 100).
currency	Order currency.
PM	Payment method.
BRAND	Card brand or similar information for other payment methods.

Possible Query Return values are:

orderID	Your order reference.
PAYID	Payment reference in our system.
PAYIDSUB	The history level ID of the maintenance operation on the PAYID.
ACCEPTANCE	Acceptance code returned by acquirer.
STATUS	Transaction status.
NCERROR	Error code.
NCSTATUS	First digit of NCERROR.
NCERRORPLUS	Explanation of the error code.
amount	Order amount (not multiplied by 100).
currency	Order currency.
PM	Payment method.
BRAND	Card brand or similar information for other payment methods.
ECI	Electronic Commerce Indicator.
CARDNO	The masked credit card number.
IP	Customer's IP address, as detected by our system in a 3-tier integration, or sent to us by the merchant in a 2-tier integration.

Eway Keys

Possible return values are:

ewayTrxnError –	This is the response returned by the bank, and can be related to both successful and failed transactions.
ewayTrxnStatus –	This value is returned to your website. For a successful transaction **"True"** is passed and for a failed transaction **"False"** is passed. This allows you to make your own logging on your website for transaction status.
ewayTrxnNumber –	This value is returned to your website. You can pass a unique transaction number from your site. You can update and track the status of a transaction when eWAY returns to your site. NB. This number is returned as **'ewayTrxnReference'**. The number returned as **'ewayTrxnNumber'**, is actually the unique eWAY Transaction number, created by eWAY itself.
ewayTrxnOption1 –	This value is returned to your website. An additional field for you to pass and receive information from eWAY.
ewayTrxnOption2 –	This value is returned to your website. An additional field for you to pass and receive information from eWAY.
ewayTrxnOption3 –	This value is returned to your website. An additional field for you to pass and receive information from eWAY.
ewayReturnAmount –	Can be used a check that the transaction is processed for the same amount as you request from your website.
ewayAuthCode –	If the transaction is successful, this is the bank authorisation number. This is also sent in the email receipt.
ewayTrxnReference –	See **'eWAYTrxnNumber'**.

IV. Links for using “OptParam”

Many of the functions allow you to append key value pairs by using the “OptParam.” This optional parameter is an advanced function and will require that you reference the following guides below depending on your gateway provider.

Advanced Integration Method (AIM) Developer Guide provided by Authorize.net:
http://www.productivecomputing.com/docs/docs_library/FM_CreditCard/AIM_guide.pdf

PP_PayflowPro_Guide provided by PayPal:
http://www.productivecomputing.com/docs/docs_library/FM_CreditCard/PP_PayflowPro_Guide.pdf

Ogone_DirectLink Guide provided by Ogone:
http://www.productivecomputing.com/docs/docs_library/FM_CreditCard/Ogone_DirectLink_EN.pdf

V. Contact Us

Successful integration of a FileMaker plug-in requires the creation of integration scripts within your FileMaker solution. A working knowledge of FileMaker Pro, especially in the areas of scripting and calculations is necessary. If you need additional support for scripting, customization or setup (excluding registration) after reviewing the videos, documentation, FileMaker demo and sample scripts, then please contact us via the avenues listed below.

Phone: 760-510-1200
Email: support@productivecomputing.com
Forum: www.productivecomputing.com/forum

Please note assisting you with implementing this plug-in (excluding registration) is billable at our standard hourly rate. We bill on a time and materials basis billing only for the time in minutes it takes to assist you. We will be happy to create your integration scripts for you and can provide you with a free estimate if you fill out a Request For Quote (RFQ) at www.productivecomputing.com/rfq. We are ready to assist and look forward to hearing from you!