



Change Log

March, 2020 – Added the RecurringCustomerEdit option.

March, 2020 – Added RecurrID to ILF Single Transaction response.

September, 2020 – Added / Changed the post back urls to be sent in the authentication rather than in the ILF frame.

September, 2020 – Added ability for multiple stored cards for a single customer.

January, 2022 –

- * Added Version 2.
- * Removed the postback urls from authentication routine as it is now all in the extravars2.
- * Version 2 is more secure than version 1 because it POSTS almost all the form fields instead of GET.
- * Added the option in ILF processing setting to be able to use 1 of 3 Customer Edit styles.



Inline Frames API V2

Last Update: January 11, 2022

Purpose: to be able to use an inline frame to process transactions, update SecureVault customers or update a recurring transaction without the need to ever touch, transmit or store any cardholders (credit card) data on the merchant's Web site or have any such data pass through the merchant's servers or network. The primary benefit is a substantially streamlined process for PCI compliance.

NOTE: This API must be enabled in the Processing Settings under Inline Frames API.

To make things easier for integration we have a PHP script with all the functions you need. There are also several examples to look at.

Version 1: Please download https://www.quantumgateway.com/files/QGW_ILF.zip

Version 2: Please download https://www.quantumgateway.com/files/QGW_ILF_V2.zip

There are 5 methods that can be performed with this API Process Transaction

Edit A Customer

Edit A Customer's Profile Only

Edit A Customer's Payment Information Only

Edit A Recurring Customer

In version 2, there is another option. It does not use method, but RequestType. See PostVars below.

Authentication

Before using any of our options/methods below, you must authenticate your session.

Type: POST

URL: https://secure.quantumgateway.com/cgi/ilf_authenticate.php

Item Name	Field Name	Values	Notes
API Username	API_Username	varchar	Randomly generated username from the Process Settings
API Key	API_Key	varchar	Randomly generated key from the Process Settings
Random Int	randval	int	Sets a unique random integer between 1111111111 and 9999999999
Ip Address	lastip	varchar	Customer's IP Address
Version	v	int	Empty or 1 for original ILF. 2 for Version 2.

Example from inside the `quantumilf_getCode` function from our `quantum_iframe.php` script.

```
<?php
    $random = rand(1111111111, 9999999999);
    $random = (int)$random;
    $version=2;
    $response = _quantumilf_http_post("secure.quantumgateway.com",
"/cgi/ilf_authenticate.php",
array (
    'API_Username' => $API_Username,
    'API_Key' => $API_Key,
    'randval' => $random,
    'lastip' => $_SERVER['REMOTE_ADDR'],
    'v' => $version,
), 443
);

print_r($response, 1);

?>
```



Refresh

Before using any of our options/method below, you must first authenticate your session.

Purpose: To help keep your customer's session from timing out using ajax. See quantum_iframe.php for example.

Type: POST

URL: https://secure.quantumgateway.com/cgi/ilf_refresh.php

Item Name	Field Name	Values	Notes
k	k	varchar	Returned value from Authentication Version 2 contains a in this response. k is before the and c1 is after it.
ip	ip	varchar	Customer IPAddress Version 1 only
c1	c1	int	Returned value from Authentication Version 2 contains a in this response. k is before the and c1 is after it. Required for Version 2
Version	v	int	Empty or 1 for original ILF. 2 for Version 2. Required for Version 2

PostVars

Type: POST

URL: <https://secure.quantumgateway.com/cgi/ilf.php>

Version 2 – To help with security, we have adapted our API to post all of the extra options/data to our API instead of passing them along inside the <iframe></iframe> source code.

All items besides v, k, c1, height and width should be included in this.

Below is part of what we have in our quantumilf_getCode function in the quantum_iframe.php

<?php

```
$extrapars = "";

if ($method != '0') {
    $extrapars .= "&METHOD=$method";
}

$extrapars.=$extrapars2;

// Version 2 posts the extrapars using the authenticated information. When the frame is generated,
then that data will be retrieved IN the frame.

if(empty($extrapars)) {} else {

$LOGINURL = "https://secure.quantumgateway.com/cgi/ilf.php";
$agent = $_SERVER['HTTP_USER_AGENT'];

$POSTFIELDS =
"&RequestType=PostVars".
"&v=".$version.
"&c1=".$c1.
"&k=".$k.

$extrapars;

$ch = curl_init();
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION,1);
Quantum Gateway ILF API
```



```
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_USERAGENT, $agent);
curl_setopt($ch, CURLOPT_URL,$LOGINURL);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,$POSTFIELDS);
curl_setopt($ch, CURLOPT_SSLVERSION, 6);
$result_curl = curl_exec ($ch);
curl_close ($ch);
$mytest = $result_curl;
}

//End Get extra vars to post/update the ILF query
```

?>

Process Transaction

Type: POST

URL: <https://secure.quantumgateway.com/cgi/ilf.php>

Minimum Requirements

Item Name	Field Name	Values	Notes
Amount	Amount	Currency	Currency Format. X.yy
ID	ID	varchar	The ID can be anything you wish. ID, email address, username, etc. Something that will get updated when they click on Finish.

Options

Item Name	Field Name	Values	Notes
Customer ID	CustomerID	varchar	Some ID you know your customer by in your own system.
Add To Vault	AddToVault	Y/N	If set to Y this option will then add your customer to your vault after a successful AUTH or SALE

Note: To use AddToVault, you must also set the CustomerID field.

Billing Information

First Name	FNAME	varchar	
Last Name	LNAME	varchar	
Address	BADDR1	varchar	
City	BCITY	varchar	
State	BSTATE	varchar	
Zipcode	BZIP1	varchar	
Country	BCOUNTRY	varchar	
Customer Email	BCUST_EMAIL	varchar	

Shipping Information

Skip Shipping Info	skip_shipping_info	Y/N	If Y then shipping information will not be shown during payment.
First Name	SFNAME	varchar	
Last Name	SLNAME	varchar	
Address	SADDR1	varchar	
City	SCITY	varchar	

State	SSTATE	varchar	
Zipcode	SZIP1	varchar	
Country	SCOUNTRY	varchar	
Override Gateway Emails			
Merchant Receipt	override_trans_email	Y/N	Y means send
Customer Receipt	override_email_customer	Y/N	Y means send
Return URLs (Optional) – Overrides default gateway settings			
Approved	post_return_url_approved	Used with Processing a Sale / Transaction	
Declined	post_return_url_declined		
Post Back URL	post_back_url	Used for Vault Or Recurring Update	
Recurring			
	RID	int	Recipe ID
	recur_times	int	Times to recur
	OverRideRecureDay	Y/N	Set recur day to current date
	override_recur	Y/N	Override the default recur amount
	RecurrID	int	Recurring transaction ID
More Options			
Phone	phone	varchar	
Invoice Number	invoice_num	varchar	
Transaction Type	trans_type	varchar	AUTH_ONLY, SALES, AUTH_CONVERT EFT – DEBIT or CREDIT(Return)
Transaction Method	trans_method	char	CC or EFT
	MAXMIND	int	1 – Use Maxmind 2 – Don't use Maxmind
Stored Card ID	SCID	int	See notes 1 below.
	PayInfoOnly	Y/N	If Y, then only the credit card information (or EFT if enabled) will be shown/asked for.
	name_on_card	varchar	some options, this is shown instead of first and last name
	item_description	varchar	
Required Information – Force customers to enter this information			
First Name	Require_FNAME	Y/N	
Last Name	Require_LNAME	Y/N	
Phone	Require_phone	Y/N	
City	Require_BCITY	Y/N	
State	Require_BSTATE	Y/N	
Address	Require_BADDR1	Y/N	
Zip	Require_BZIP1	Y/N	
Country	Require_BCOUNTRY	Y/N	

CVV2	Require CVV2	Y/N	
------	--------------	-----	--

Note 1: SCID (Stored Card ID) – This is used in conjunction with AddToVault. A CustomerID can have multiple cards on file if the SCID is used. SCID can then also be used with the customer edit options.

extrapars2

In order to make our ILF more customizable, we have limited the below iframe function down to its minimum and have added an \$extrapars2.

\$extrapars2 can be added to outside of the quantumilf_getCode function and passed to it.

It can be used just like items in the quantumilf_getCode.

Example:

```
if(empty($country)) {} else {  
    $extrapars2 .= "&BCOUNTRY=$country";  
}
```

Remember this has to be done before you call the quantumilf_getCode.

Required/Minimum are

- * API_Username
- * API_Key
- * width
- * height
- * method

Everything else, including Amount, id and custid should be included in your \$extrapars2.

post_return_url_approved, post_return_url_declined and post_back_url should also be included in your \$extrapars2 as the approved and declined are not needed for editing a customer and post_back_url is not needed for processing a transaction.

CustomerEdit

- * API_Username
- * API_Key
- * CustomerID
- * Method – CustomerEdit

Sample - Full options – Vars preset

```
<?php
```

```
$width="800";
```

```
$height="800";
```

```
$custid="1639410668";
```

```
$method='CustomerEdit';
```

```
$post_back_url="https://mywebsite.com/approved.php";
```

```
$extrapars2 .= "&METHOD=$method";
```

```
$extrapars2 .= "&CustomerID=" . urlencode($custid);
```

```
$extrapars2 .= "&post_back_url=$post_back_url";
```

```
$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,  
$extrapars2);
```

```
?>
```

- - Required

Minimum Requirements

Item Name	Field Name	Values	Notes
Customer ID	CustomerID	CustomerID	ID that you know your customer by.

Options

Item Name	Field Name	Values	Notes
First Name	FNAME	varchar	
Last Name	LNAME	varchar	
Address	BADDR1	varchar	
City	BCITY	varchar	
State	BSTATE	varchar	
Zipcode	BZIP1	varchar	
Country	BCOUNTRY	varchar	
Customer Email	BCUST_EMAIL	varchar	
Phone	phone	varchar	
Stored Card ID	SCID	int	

CustomerEditProfile

- * API_Username
- * API_Key
- * CustomerID
- * Method – CustomerEdit

Sample - Full options – Vars preset

```
<?php
```

```
$width="800";
$height="800";
$custid="1639410668";
$method='CustomerEditProfile';
$post_back_url="https://mywebsite.com/approved.php";
```

```
$extrapars2 .= "&METHOD=$method";
$extrapars2 .= "&CustomerID=" . urlencode($custid);
$extrapars2 .= "&post_back_url=$post_back_url";
```

```
$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,
$extrapars2);
```

```
?>
```

- - Required

Minimum Requirements

Item Name	Field Name	Values	Notes
Customer ID	CustomerID	CustomerID	ID that you know your customer by.

Options

Item Name	Field Name	Values	Notes
First Name	FNAME	varchar	
Last Name	LNAME	varchar	
Address	BADDR1	varchar	
City	BCITY	varchar	
State	BSTATE	varchar	
Zipcode	BZIP1	varchar	
Country	BCOUNTRY	varchar	
Customer Email	BCUST_EMAIL	varchar	
Phone	phone	varchar	
Stored Card ID	SCID	int	



CustomerEditPayment

- * API_Username
- * API_Key
- * CustomerID
- * Method – CustomerEditPayment

Sample

```
<?php
```

```
$width="800";
```

```
$height="800";
```

```
$custid="1639410668";
```

```
$method='CustomerEditPayment';
```

```
$post_back_url="https://mywebsite.com/approved.php";
```

```
$extrapars2 .= "&METHOD=$method";
```

```
$extrapars2 .= "&CustomerID=" . urlencode($custid);
```

```
$extrapars2 .= "&post_back_url=$post_back_url";
```

```
$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,  
$extrapars2);
```

```
?>
```

* - Required

This option will only show the update payment information.

Minimum Requirements

Item Name	Field Name	Values	Notes
Customer ID	CustomerID	CustomerID	ID that you know your customer by.

Options

Stored Card ID	SCID	int	
----------------	------	-----	--

RecurringCustomerEdit

- * API_Username
- * API_Key
- * RecurrID
- * Method – RecurringCustomerEdit

Sample

```
<?php
```

```
$width="800";
```

```
$height="800";
```

```
$custid="";
```

```
$method='RecurringCustomerEdit';
```

```
$RecurrID="26117";
```

```
$post_back_url="https://mywebsite.com/approved.php";
```

```
$extrapars2 .= "&METHOD=$method";
```

```
$extrapars2 .= "&RecurrID=$RecurrID";
```

```
$extrapars2 .= "&post_back_url=$post_back_url";
```

```
$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,  
$extrapars2);
```

```
?>
```

* - Required

This option will only show the update the billing and payment information.

Sample 1 - Minimum options only for processing

```
<?php
$width="800";
$height="800";
$method="";
$extrapars2 .= "&ID=" . $id;
$extrapars2 .= "&Amount=$amount";

$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,
$extrapars2);
?>
```

Sample 2 - AddToVault option, but missing CustomerID. This sample will not add the customer to the vault.

```
<?php
$width="800";
$height="800";
$method="";
$extrapars2 .= "&ID=" . $id;
$extrapars2 .= "&Amount=$amount";
$extrapars2 .= "&AddToVault=Y";

$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,
$extrapars2);
?>
```

Sample 3 - AddToVault option and the skip_shipping_info option

```
<?php
$width="800";
$height="800";
$method="";
$extrapars2 .= "&ID=" . $id;
$extrapars2 .= "&Amount=$amount";
$extrapars2 .= "&AddToVault=Y";
$extrapars2 .= "&CustomerID=" . urlencode($custid);
$extrapars2 .= "&skip_shipping_info=Y";

$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,
$extrapars2);
?>
```

Sample 4 – Full options – Vars preset

<?php

For this example, we will only add items to the extrapars2 if it exists.

```
$width="800";
```

```
$height="800";
```

```
$method="";
```

```
$extrapars2="";
```

```
    if ($addtoVault == 'Y') {
```

```
        $extrapars2 .= "&AddToVault=Y";
```

```
    }
```

```
    if ($custid != '0') {
```

```
        $extrapars2 .= "&CustomerID=" . urlencode($custid);
```

```
    }
```

```
    if ($amount > '0') {
```

```
        $extrapars2 .= "&Amount=$amount";
```

```
    }
```

```
    if ($id > '0')    {//ID is a must for processing.
```

```
        $extrapars2 .= "&ID=" . $id;
```

```
    }
```

```
    if(empty($fname)) {} else {
```

```
        $extrapars2 .= "&FNAME=$fname";
```

```
    }
```

```
    if(empty($lname)) {} else {
```

```
        $extrapars2 .= "&LNAME=$lname";
```

```
    }
```

```
    if(empty($address)) {} else {
```

```
        $extrapars2 .= "&BADDR1=$address";
```

```
    }
```

```
    if(empty($city)) {} else {
```

```
        $extrapars2 .= "&BCITY=$city";
```

```
    }
```

```
    if(empty($state)) {} else {
```

```
        $extrapars2 .= "&BSTATE=$state";
```

```
    }
```

```
    if(empty($zip)) {} else {
```

```
        $extrapars2 .= "&BZIP1=$zip";
```

```
    }
```

```
    if(empty($country)) {} else {
```

```
        $extrapars2 .= "&BCOUNTRY=$country";
```




```
}
if(empty($phone)) {} else {
    $extrapars2 .= "&phone=$phone";
}
if(empty($email)) {} else {
    $extrapars2 .= "&BCUST_EMAIL=$email";
}
if(empty($invoice_num)) {} else {
    $extrapars2 .= "&invoice_num=$invoice_num";
}
//start shipping
if ($skipshipping == 'Y') {
    $extrapars2 .= "&skip_shipping_info=Y";
}
if(empty($sfname)) {} else {
    $extrapars2 .= "&SFNAME=$sfname";
}
if(empty($slname)) {} else {
    $extrapars2 .= "&SLNAME=$slname";
}
if(empty($saddress)) {} else {
    $extrapars2 .= "&SADDR1=$saddress";
}
if(empty($scity)) {} else {
    $extrapars2 .= "&SCITY=$scity";
}
if(empty($sstate)) {} else {
    $extrapars2 .= "&SSTATE=$sstate";
}
if(empty($szip)) {} else {
    $extrapars2 .= "&SZIP1=$szip";
}
if(empty($scountry)) {} else {
    $extrapars2 .= "&SCOUNTRY=$scountry";
}
// end shipping
if(empty($strans_type)) {} else {
    if($strans_type=='AUTH_ONLY' || $strans_type=='AUTH_CAPTURE' ||
$strans_type=='SALES') {
        $extrapars2 .= "&trans_type=$strans_type";
    }
}
if(empty($Vendor)) {} else {
    $extrapars2 .= "&Vendor=$Vendor";
}
}
//Y to send. No to not send
Quantum Gateway ILF API
```



```
if($override_trans_email=='Y' || $override_trans_email=='N') {
    $extrapars2 .= "&override_trans_email=$override_trans_email";
}
//Y to send. No to not send
if($override_email_customer=='Y' || $override_email_customer=='N') {
    $extrapars2 .= "&override_email_customer=$override_email_customer";
}

//start recurring
if($RID>0) {
    $extrapars2 .= "&RID=$RID";
}

if($OverRideRecureDay=='Y') {
    $extrapars2 .= "&OverRideRecureDay=$OverRideRecureDay";
}

if($override_recur=='Y') {
    $extrapars2 .= "&override_recur=$override_recur";
}

if($recur_times>0) {
    $extrapars2 .= "&recur_times=$recur_times";
}
if($RecurrID>0) { // usedd for RecurringCustomerEdit
    $extrapars2 .= "&RecurrID=$RecurrID";
}

//end recurring
if($PayInfoOnly=='Y') {
    $extrapars2 .= "&PayInfoOnly=Y";
}

if(empty($item_description)) {} else {
    $extrapars2 .= "&item_description=$item_description";
}

if($Require_CVV2=='Y') {
    $extrapars2 .= "&Require_CVV2=Y";
}

if(empty($post_return_url_approved)) {} else {
    $extrapars2 .= "&post_return_url_approved=$post_return_url_approved";
}
}
```



```
if(empty($post_return_url_declined)) {} else {  
    $extrapars2 .= "&post_return_url_declined=$post_return_url_declined";  
}
```

```
$quantum = quantumilf_getCode($API_UNAME, $API_KEY, $width, $height, $method,  
$extrapars2);
```

```
?>
```



Possible URL Responses

CustomerUpdate/UpdateCustomer

<https://www.yourdomain.com/UpdateCustomer.php?Status=APPROVED&CustomerID=8675309>

<https://www.yourdomain.com/UpdateCustomer.php?Status=DECLINED&CustomerID=8675309>

Processing a transaction

https://www.yourdomain.com/Approved.php?trans_result=APPROVED&transID=888888&ID=8675309

https://www.yourdomain.com/Declined.php?trans_result=DECLINED&transID=888888&ID=8675309&decline_reason=AUTH_DECLINE&errorcode=200

Please note that when using URL Responses you will need to use GET or REQUEST to get the results.



What can I customize?

In the processing settings you can set the following.

- **Background Color.**
- **Font size.**
- **Font Family.**
- **Font color.**
- **Form field font color**
- **Form field font size**
- **Form field background color**
- **Company Logo** to be used on the Process a Transaction option.
- **Page Heading** to be used on all options above.
- **Payment Heading** to be used on the Process a Transaction option.
- **Approved URL** to be used with Process a Transaction and CreateTransaction.
- **Declined URL** to be used with Process a Transaction and CreateTransaction.
- **Payment Update URL** (\$post_back_url) to be used with CustomerEdit/Update



Version 2.0 POSTs the extravars data instead of adding it into the iframe. This makes the form more secure as your customer can't see all of the vars.

V1/V2 Examples

V1

<iframe

```
src="https://secure.quantumgateway.com/cgi/ilf.php?k=*****&ip=127.0
.0.1&METHOD=&AddToVault=Y&CustomerID=1639403291&Amount=15.00&ID=16394031
80&FNAME=John&LNAME=Doe&BADDR1=1515 15th
St&BCITY=Chesapeake&BSTATE=VA&BZIP1=23320&BCOUNTRY=USA&phone=888555
1212&BCUST_EMAIL=john.doe@somedomain.com&invoice_num=CD1212&SFNAME=John
&SLNAME=Doe&SADDR1=1515 15th
St&SCITY=Chesapeake&SSTATE=VA&SZIP1=23320&SCOUNTRY=USA&trans_type=AUT
H_CAPTURE&post_return_url_approved=https://mydomain.com/ilf_approved.php&post_retur
n_url_declined=https://mydomain.com/ilf_declined.php&name_on_card=John
Doe&override_trans_email=N&override_email_customer=N" height="800" width="800"
frameborder="0"></iframe>
```

V2

<iframe

```
src="https://secure.quantumgateway.com/cgi/ilf.php?v=2&k=*****&c1
=****" height="800" width="800" frameborder="0"></iframe>
```