



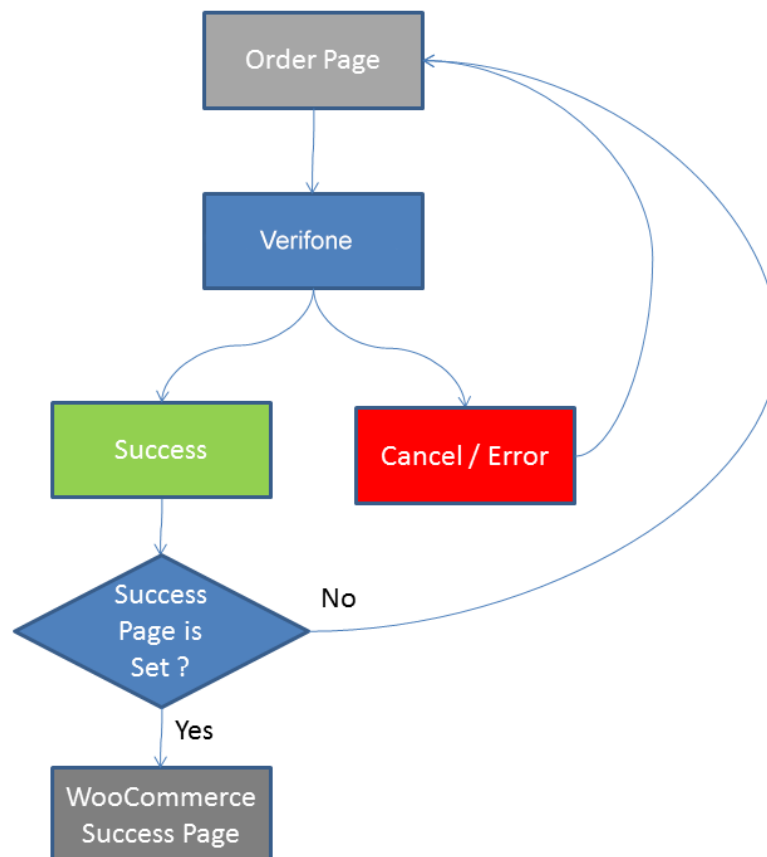
## Verifone payment plugin for WooCommerce

### *Technical specification*

This document describes technically the Verifone payment interface to enable third party web store integrations to the payment system.

### **Process**

The payment process starts at the Shop Order Page where the order is summarized and a button for moving to the Verifone payment gateway is present. Pushing the payment button posts the Verifone Payment Initialization Form to the payment gateway. The payment process ends with posting a response from the Verifone payment gateway to the WooCommerce plugin. It is possible for the user to move with a single click from the payment page to the order page to retry payment.



Verifone sends a response to the Verifone WooCommerce checkout plugin. Based on received information we determine the status of the order. In case of an error, the user is redirected to the order page. He is informed of the common error. If we receive a positive message, the plugin verifies the generated signature based on received data. If the signature is wrong, the error "Security Error. Illegal access detected." is displayed. If the signature is correct, the order is updated as completed.

The return page can be set in the plugin settings. Users will be redirected there after successful payment. If no return page is set, the user will be redirected to the order details page.

#### Plugin files

File	Description
index.php	Main plugin file with WC_Verifone_Payment class. File is responsible for all payment logic, WooCommerce settings, forms and fields.
verifone-key-pair-generator.jar	Verifone key pair generator. Requires Java to work.
images/logo.png	This Verifone logo is displayed next to the payment method on the checkout page.
keys/pay-page-public-key.pem	Demo public key for the test environment with demo-merchant-agreement Merchant ID.
keys/shop-private-key.pem	Demo private key for the test environment with demo-merchant-agreement Merchant ID.
Crypt/* Math/*	Helper files, used to generate and verify the signature.

#### Plugin functions

Function	Description
----------	-------------

init_form_fields	Defines all plugin fields, called in constructor.
admin_options	Output settings in the correct format, called in constructor.
generate_verifone_form	Generate Verifone payment system form. The form is automatically submitted.
get_verifone_args	Set all necessary data for the form mentioned above.
formatparameters	Set all data to correct format. If ';' occurs in a value, it will be replaced with ';' in the signature content string.
generatesignature	Creates a signature for data. Uses Crypt/RSA.
verifysignature	Verifies the data signature. Uses Crypt/RSA.
process_payment	Process the payment and return the result.
check_verifone_response	Checks for valid Verifone payment server callback.
verifone_item_name	Limits the length of item names to 30 characters.
formatApiNumber	Format number to API format (23,5EUR => 2350)
get_languages	Get locales supported by the Verifone payment gateway.
get_url	Shuffle available URL in a production environment.
get_currency_number	Get supported currency number (ISO 4217)
get_countries	Get supported country number (ISO 3166)
get_error_message	Translate error codes to user messages.
get_pages	Get all available WordPress pages for the "Return Page" plugin configuration

	option.
woocommerce_add_verifone_payment_gateway	Add the Verifone payment gateway to WooCommerce.

## Payment process

The plugin draws and validates one of available payment system URLs. Transport between the shop system and the payment system is carried out via web browser of the buyer and secured using transport layer security (SSL/TLS) in both connections: browser to shop system and browser to payment system. Only a server side certificate granted by a public certificate authorities can be used. Messages are encoded as HTML forms with the UTF-8 character set as part of the page buyer views and transmitted with HTTP POST requests.

## Payment initialization form parameters (used in plugin)

Parameter	Name	Description
i-f-1-11_interface-version	Version	Version of the payment interface.
i-f-1-3_order-currency-code	Currency Code	Numeric currency code (ISO 4217).
i-t-1-3_delivery-address-country-code	Delivery Address Country Code	Numeric country code of the delivery address (ISO 3166).
i-t-1-4_order-vat-percentage	VAT Percentage	Value added tax with two decimal precision.
l-f-1-20_order-gross-amount	Gross Amount	Total payment amount including taxes and discount with two decimal precision.
l-f-1-20_order-net-amount	Net Amount	Total payment amount with two decimal precision.
l-f-1-20_order-vat-amount	VAT Amount	Value added tax amount with two decimal

		precision.
locale-f-2-5_payment-locale	Locale	The language locale used by buyer. Supported locales are: fi_FI, sv_SE, no_NO and en_GB.
s-f-1-100_buyer-email-address	Email Address	Email address of the buyer.
s-f-1-10_software-version	Version	Version of the payment system.
s-f-1-30_buyer-first-name	First Name	First name of the buyer.
s-f-1-30_buyer-last-name	Last Name	Last name of the buyer.
s-f-1-30_software	Software	Name of the web shop software.
s-f-1-36_merchant-agreement-code	Merchant Agreement Code	Textual code of the merchant agreement.
s-f-1-36_order-number	Order Number	Textual order number assigned by shop system.
s-f-32-32_payment-token	Payment Token	SHA-256 hash with combination of merchant agreement code, order number
s-f-5-128_cancel-url	Cancel URL	Url to Verifone plugin
s-f-5-128_error-url	Error URL	
s-f-5-128_expired-url	Expired URL	
s-f-5-128_rejected-url	Rejected URL	
s-f-5-128_success-url	Success URL	
s-t-1-30_delivery-address-city	Delivery Address City	City of the delivery address.
s-t-1-30_delivery-address-line-one	Delivery Address Line #1	Line one of the delivery address.
s-t-1-30_delivery-address-	Delivery	Line two of the delivery

line-three	Address Line #2	address.
s-t-1-30_delivery-address-line-two	Delivery Address Line #3	Line three of the delivery address.
s-t-1-30_delivery-address-postal-code	Delivery Address Postal Code	Postal code of the delivery address.
s-t-1-36_order-note	Order Note	Custom parameter.
t-f-14-19_order-timestamp	Order Timestamp	UTC timestamp defining the orders time from web shop point of view.
t-f-14-19_payment-timestamp	Payment Timestamp	UTC timestamp defining the payment start time from web shop point of view.
i-t-1-1_skip-confirmation-page	Skip Confirmation	Integer defining if confirmation page after successful payment should be skipped.
s-t-256-256_signature-one	Signature One	128 byte signature converted to uppercase hexadecimal string.

## Items

Parameter	Name	Description
s-t-1-30_bi-name-<N>	Item Name	Item name of the Nth basket item.
i-t-1-11_bi-unit-count-<N>	Item Unit Count	Number of units in the item.
i-t-1-4_bi-discount-percentage-<N>	Item Discount Percentage	Item discount percentage tax with two decimal precision.
i-t-1-4_bi-vat-percentage-<N>	Item VAT Percentage	Item value added tax percentage with two decimal precision.
l-t-1-20_bi-gross-amount-<N>	Item Gross Amount	Item gross amount including tax and discount with two decimal precision.
l-t-1-20_bi-net-amount-<N>	Item Net Amount	Item net amount calculated from unit cost

		times unit count with two decimal precision.
l-t-1-20_bi-unit-cost-<N>	Item Unit Cost	Unit cost with two decimal precision but without discount and tax.

### Page URLs

Name	Description
Verifone E-Commerce Payment Node #1 Availability	The URL to payment node #1 availability check page.
Verifone E-Commerce Payment Node #2 Availability	The URL to payment node #2 availability check page.
Verifone E-Commerce Payment Node #3 Availability	The URL to payment node #3 availability check page.
Web Shop Payment Success Page	The URL to response from Verifone. For Verifone plugin is will always be WC()->api_request_url( 'WC_Verifone' ) http://<yourdomain>/?wc-api=WC_Verifone
Web Shop Payment Rejected Page	
Web Shop Payment Canceled Page	
Web Shop Payment Expired Page	
Web Shop Payment Error Page	

### Verifone response

Order number is passed in **s-f-1-36\_order-number** parameter. If a value is set, the script tries to find the order from the database. If the order exists, the script validates the signature based on keys and sent data.

After successful signature validation, the script checks if the **s-t-1-30\_cancel-reason** parameter with an error message is set. If an error exists, the user is redirected to the order page. If **s-t-1-30\_cancel-reason** is not set, payment was successful.

On successful processing of the post, the shop system will respond with HTTP status 200 OK. Any other response code indicates availability problems in the shop system.

### **Payment cancellation scenarios**

The message value is passed in the cancel result posts (using parameter **s-t-1-30\_cancel-reason**).

<b>Error</b>	<b>Description</b>
cancel-payment-rejected	Payment was rejected by payment acquirer due to restrictions placed on payment method or lack of funds in the corresponding account.
cancel-user-canceled	Buyer canceled to payment process.
cancel-payment-expired	The payment process expired due to timeout or usage of browser navigation buttons.
cancel-system-error	The payment process failed due to error in the payment system or payment acquirer system.
cancel-inv-payment-params	The payment process failed due to payment parameters from merchant, that lead to situation that could not be fulfilled, such as an attempt to make recurring bank payment.
cancel-external-system-error	The payment failed due to external system giving invalid message authentication code.
cancel-save-rej-not-enrolled	Attempt to save a non 3d secure enrolled card, when this is not supported by acquirer agreement for this type of card.
cancel-paym-rej-not-enrolled	Attempt to pay with non 3d secure enrolled card more than the maximum set by acquirer agreement of the merchant for that card type.



cancel-ext-address-mismatch	External system has different address than given one.
cancel-too-many-sms-passwords	User tried to enter wrong sms password too many times.
cancel-too-many-input-retries	Page has been reloaded for too many times. Browsers have been instructed not to reload payment web pages, so if this happens payment web assumed that the request is not from browser or browser is reloading page infinitely.
cancel-bank_axess_error	Payment rejected because of a BankAxess error code returned. (Example: J1: Bank does not support BankAxess).

### Key file handling

The customer generates a 1024 bit long RSA private key - public key pair (or self signed X.509 certificate) and delivers their public key in PEM format (or certificate in .cer format) to Verifone via TCS Client.

```
java -jar verifone-key-pair-generator.jar
```

The command creates two files:

- *your-merchant-agreement-code-private.pem*
- *your-merchant-agreement-code-public.pem*

Where *your-merchant-agreement-code* is your web store identifier issued by Verifone (ECxxxxxxx). Once generated, the private key must be stored on the server. The public key must be delivered to Verifone via TCS reporting service. You will also need the Verifone public key which is also available via TCS reporting service. See the Verifone production deployment guide for further instructions.

Once you have generated both a private and public key for your store, delivered the public key to Verifone and obtained a separate public key from Verifone, place both your generated private key and the public key obtained from Verifone into wp-content/plugins/verifone-payment-gateway-woocommerce/keys in your WordPress installation.

## Testing

It is recommended to test the implementation first in the customer test environment with demo merchant keys. The demo merchant agreement code is 'demo-merchant-agreement'. The demo keys are provided in the integration kit.

The password for the demo-merchant-agreement.p12 is 'password' (without quotes).

To enable the test environment, enable the following configuration in the settings for the Verifone WooCommerce plugin.

Configuration option	Value
Test environment	Enable
Debug Log	Enable
Merchant ID	demo-merchant-agreement
Shop private key file	keys/shop-private-key.pem
Pay page key file	keys/pay-page-public-key.pem
Test payment URL	https://epayment.test.point.fi/pw/payment
Node availability	Checked

If Debug Log is enabled, the customer can follow all responses from Verifone.

Log files are placed In Woocommerce → System Status → Log tab.

### Example of success transaction:

01-01-2015 @ 08:00:00 - WooComm - Generating payment form for order #1.

Notify URL: http://test-point.com/?wc-api=WC\_Verifone

01-01-2015 @ 08:02:00 - Api response for order - 1

01-01-2015 @ 08:02:01 - Api response for order - 1- - success - Payment recived

### Example of canceled transaction:

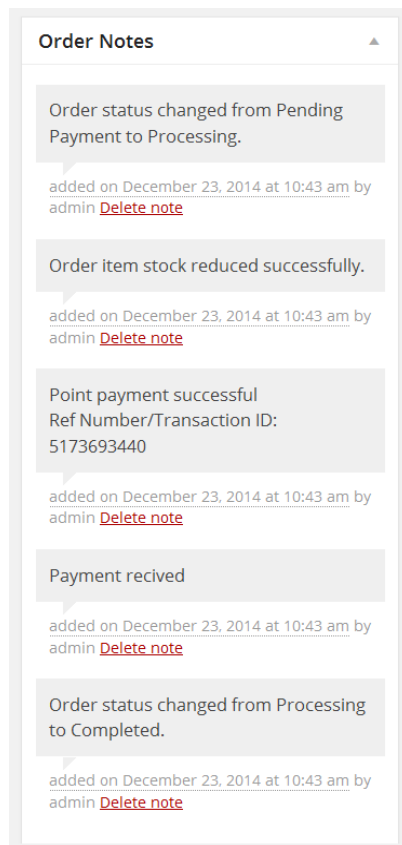
01-01-2015 @ 09:00:00 - WooComm - Generating payment form for order #2.

Notify URL: [http://test-point.com/?wc-api=WC\\_Verifone](http://test-point.com/?wc-api=WC_Verifone)

01-01-2015 @ 09:02:00 - Api response for order - 2

01-01-2015 @ 09:02:01 - Api response for order - 2 - cancel-user-canceled - error - Buyer canceled to payment process.

Additionally, the plugin saves information in order details:



### Tested operations:

Operation	Status
Making an order with one product, tax rate 15%, payment method American Express.	Redirects correctly to the order page (as configured), all prices match up (both including and excluding tax)
Making an order with multiple products, tax rate 15%, payment method American Express.	Redirects correctly to the order page (as configured), all prices match up (both including and excluding tax)
Making an order with multiple products, tax rates ranging from 0% to 24% percent, payment method	Redirects correctly to the order page (as configured), all prices match up (both including and excluding tax)

American Express.	
Proceeding to the Verifone interface and returning back to the store without paying.	Redirects back to the checkout page where an error is shown (as expected). Order is not completed.
Turning on "Enable auto return"	After successful payment redirects back to the checkout page. Order is completed.
Turning on "Enable auto return" and setting return page.	After successful payment redirects correctly to the selected page. Order is completed.
Turning off "Enable auto return". In the Verifone system in eCommerce -> Merchant settings, set Shop success URL "http://<yourdomain>/?wc-api=WC_Verifone_Payment"	After successful payment stay on Point system. Close browser without returning to shop. Order is completed.
Turning on "Enable auto return". In the Verifone system in eCommerce -> Merchant setting, set Shop success URL "http://<yourdomain>/?wc-api=WC_Verifone_Payment"	After successful payment redirects back to the checkout page. Order is completed.

## Production configuration options

Configuration option	Value
Test environment	Disable
Debug Log	Enable (not necessary)
Merchant ID	YOUR MERCHANT ID
Shop private key file	keys/YOUR-GENERATED-PRIVATE-KEY.pem
Pay page key file	keys/YOUR-GENERATED-PUBLIC-KEY.pem
Required. URL to the payment system	https://epayment1.point.fi/pw/payment
Optional. Second URL to the payment system	https://epayment2.point.fi/pw/payment

Optional. Third URL to the payment system	<a href="https://epayment3.point.fi/pw/payment">https://epayment3.point.fi/pw/payment</a>
Node availability	Checked