



Verifone Payment extension for Magento eCommerce platform

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

Table of contents

1. Introduction
2. Payment process
3. Extension files
4. Payment initialization form parameters
5. Verifone gateway responses
6. Default callback URLs
7. Payment cancellation scenarios
8. Key file handling
9. Testing
10. Admin panel configuration options
11. Cron job settings

1. Introduction

The purpose of this specification is to describe how Verifone payment system (later on "Verifone") is integrated to a web store (later on "store") based on the Magento ecommerce system.

2. Payment process

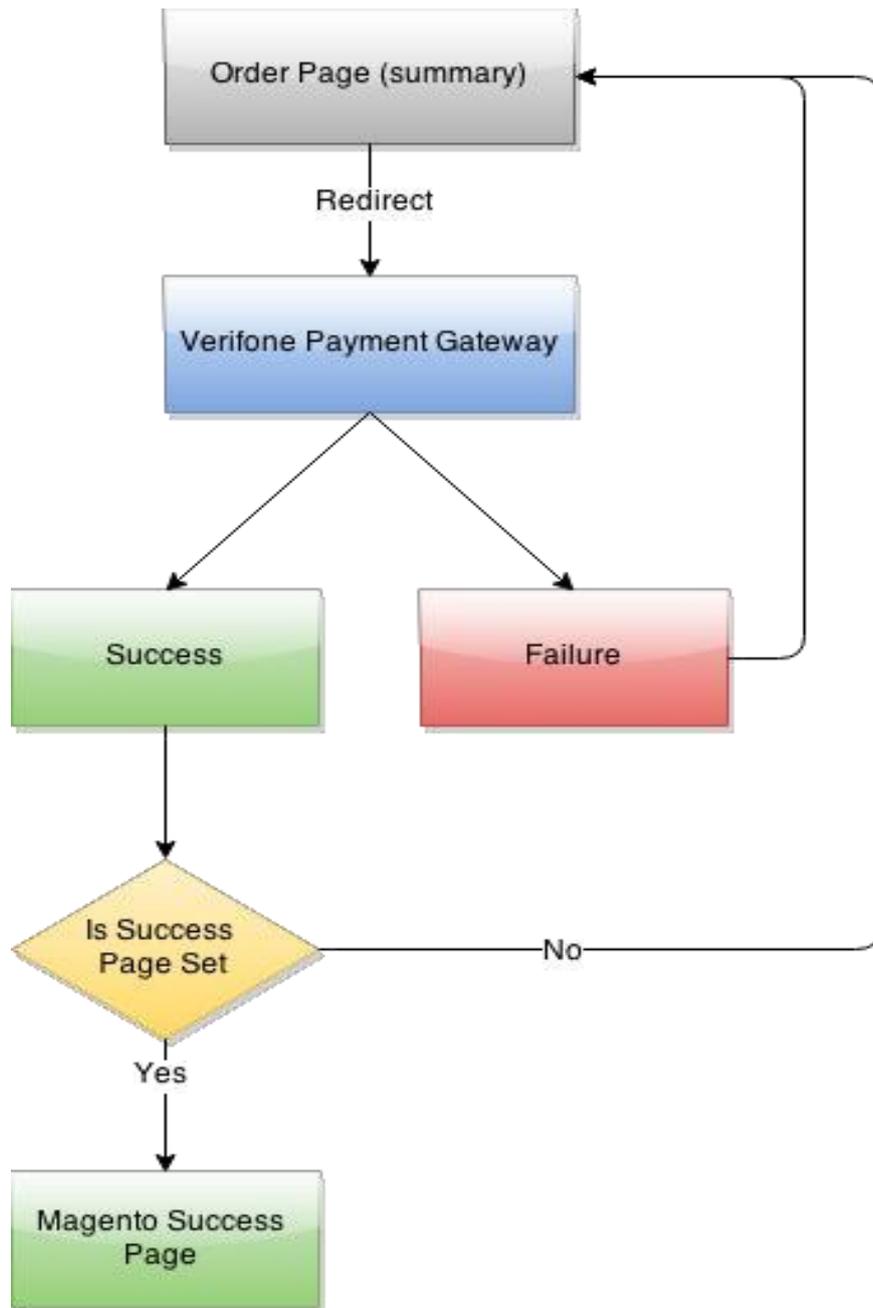
The payment process starts at the Shop Order Page where the order is summarized.

A button for redirecting 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 Magento 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 Magento 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 and is informed about the common error. If a success response is received, 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.

In Admin > System > Sales > Payment Methods > VerifonePayment > Select payment methods we may select all separated Verifone payment methods which are enabled to particular merchant. Option "VerifonePayment – All in one" will redirect the customer to default Verifone Gate with selection of all methods. Every other methods will take the customer directly to the bank or card form, and these options are displayed as default Magento separate payment methods on Magento checkout page.



3. Extension files

Extension files are located in `/app/code/community/Verifone/VerifonePayment/`

<code>etc/config.xml</code>	Main extension configuration file.
<code>etc/system.xml</code>	Payment method form in admin.
<code>Model/Payment.php</code>	Main Payment Model. File responsible for regular payments logic.
<code>Model/Payment/Recurring.php</code>	Payment Model for recurring Payments. It extends Main Payment Model. File responsible for recurring payments logic.
<code>Model/Payment/Methods/*.php</code>	Payments for all Verifone gate payments. All of them extend <code>Model/Payment.php</code> only code is different for each payment.
<code>Model/Payment/Methods/Recurring/*.php</code>	Payments for cards payments of Verifone gate payments. All of them extend <code>Model/Payment/Recurring.php</code> only code is different for each payment.
<code>Model/Db.php</code> <code>Morel/Resourse/Db.php</code> <code>Morel/Resourse/Db/Collection.php</code>	Model responsible for database. It is used to store payment method codes and to set them to be active / not active. Also responsible for queries for cron jobs (in recurring orders)
<code>Model/PIGate.php</code>	Model responsible for Payment Inteface Gate logic

Model/S2SGate.php	Model responsible for Server To Server Gate logic
Model/Observer.php	Cron jobs logic. Also cooperates with payment logic.
controllers/RedirectController.php	Main extension controller with Verifone_VerifonePayment_RedirectController class. In this file all of the actions used in payment process are declared.
controllers/Adminhtml/RefreshController.php	Controller responsible for getting available payment methods from Verifone Server To Server Gate.
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.
Helper/Cryptutils.php Helper/Crypt/* Helper/Math/*	Helper files, used to generate and verify the signature.
Helper/Data.php	Helper to get countries codes and currency codes
Helper/Forms.php	Helper which is responsible for generating all forms for both Verifone gates
Block/Redirect.php	Block Responsible to form POST form for Payment Interface Gate
Block/Payment/Methods/*	Blocks which are responsible for payments

Block/Catalog/Product/Price.php	Block which is responsible for '/' [period]' text in price rendering, [period] = [day/s, week/s, month/s, year/s] (based on Magento setting)
Block/Adminhtml/Refresh.php	Block which is responsible for refreshing available payment methods in payment setting (Admin > System > Configuration > Sales > Payment Methods).
sql/verifonepayment_setup/*	installation part – database + attributes

Extension files are located in **/app/locale/**

en_US/Verifone_VerifonePayment.csv fi_FI/Verifone_VerifonePayment.csv sv_SE/Verifone_VerifonePayment.csv	Translation files
en_US/template/email/verifone_expire_card.html en_US/template/email/verifone_recurring_payment_failed.html	Default email templates

Extension files are located in **/app/design/**

frontend/base/default/template/VerifonePayment/*	frontend checkout blocks templates – view part (MVC)
adminhtml/default/default/template/VerifonePayment/refresh.phtml	adminhtml button „Get Available Payment methods”

4. Payment initialization form parameters

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.

I-f-1-20_order-gross-amount	Gross Amount	Total payment amount including taxes and discount with two decimal precision.
I-f-1-20_order-net-amount	Net Amount	Total payment amount with two decimal precision.
I-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	Url to Verifone plugin
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-line-three	Delivery Address Line #2	Line two of the delivery 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.

I-t-1-20_bi-gross-amount- <N>	Item Gross Amount	Item gross amount including tax and discount with two decimal precision.
I-t-1-20_bi-net-amount- <N>	Item Net Amount	Item net amount calculated from unit cost times unit count with two decimal precision.
I-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.

Recurring payment

Parameter	Name	Description
i-t-1-1_recurring-payment	Recurring Payment	Recurring Payment Flag (0/1) – (not recurring / recurring)
s-t-1-30_recurring-payment-subscription-name	Recurring Payment Subscription Name	Payment Name ('FIRST NAME' 'LAST NAME')
s-t-1-30_recurring-payment-subscription-code	Recurring Payment Subscription Code	Subscription unique code
i-t-1-3_recurring-payment-subscription-expected-period	Recurring Payment Subscription Expired Period	7-999 Number of days expected between recurring payments(cycle). Safeguards payer from being billed multiple times in short period.
i-t-1-1_save-payment-method	Save Payment Method	Needs to be '1' if recurring payment is used

5. Verifone gateway responses

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.

6. Default callback URLs

Callback URLs are defined directly in code (model located in **etc/config.xml**):

success URL: *<yourdomain.com>/VerifonePayment/redirect/success/*

rejected URL: *<yourdomain.com>/VerifonePayment/redirect/rejected/*

cancel URL: *<yourdomain.com>/VerifonePayment/redirect/cancel/*

expired URL: *<yourdomain.com>/VerifonePayment/redirect/expired/*

error URL: *<yourdomain.com>/VerifonePayment/redirect/error/*

7. Payment cancellation scenarios

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

Error	Description
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	They 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 that 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).

8.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:

```
<your-magento-root-folder>/keys
```

9. 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

The test payment gateway url is:

<https://epayment-test.point.fi/pw/payment>

10. Admin panel configuration options

Verifone payment configuration options are located in:

System->Configuration->Payment Methods->VerifonePayment

Test environment

Configuration option	Description
Title	Verifone Payment
Enabled	Yes
Verifone Payment merchant agreement code	demo-merchant-agreement
Shop private key filename	keys/demo-merchant-agreement- private.pem
Verifone public key filename	keys/point-e-commerce-test-public- key.pem
Pay page URL 1	https://epayment-test.point.fi/pw/payment
Pay page URL 2	https://epayment-test.point.fi/pw/payment

Pay page URL 3	https://epayment-test.point.fi/pw/payment
Check payment node availability	No
Skip confirmation page	Yes
Payment Applicable From	All Allowed Countries
Countries Payment Applicable From	No selection
Sort order	1
Select payment methods	Select needed option.
Get Available Payment methods	Click to fill „Select payment methods” list
Add period label to price	No
Recurring payment failed email notify	Default Template from Locale

Production environment

Configuration option	Description
Title	Verifone Payment
Enabled	Yes
Verifone Payment merchant agreement code	YOUR MERCHANT ID
Shop private key filename	keys/YOUR-GENERATED-PRIVATE- KEY.pem
Verifone public key filename	keys/ YOUR-GENERATED-PUBLIC- KEY.pem
Pay page URL 1	https://epayment1.point.fi/pw/payment
Pay page URL 2	https://epayment2.point.fi/pw/payment
Pay page URL 3	https://epayment3.point.fi/pw/payment
Check payment node availability	Yes
Skip confirmation page	No
Payment Applicable From	Select needed option
Countries Payment Applicable From	Select needed countries

Sort order	Set needed sort order
Select payment methods	Select needed option.
Get Available Payment methods	Click to fill „Select payment methods” list
Add period label to price	No
Recurring payment failed email notify	Default Template from Locale

11. Cron job settings in file etc/config.xml

Charge process:

```
<crontab>
  <jobs>
    <VerifonePayment_charge>
      <schedule>
        <cron_expr>0 1 * * *</cron_expr>
      </schedule>
      <run>
        <model>VerifonePayment/observer::chargeRecurringProfiles</model>
      </run>
    </VerifonePayment_charge>
  </jobs>
</crontab>
```

this means that method chargeRecurringProfiles will be executed everyday on 01:00:00, If Payment Profile is ready to charge then a charge request will be executed.

Check for expire process:

```
<crontab>
  <jobs>
    <VerifonePayment_charge>
      <schedule>
        <cron_expr>0 1 * * *</cron_expr>
      </schedule>
      <run>
        <model>VerifonePayment/observer::checkIfCardIsExpired</model>
      </run>
    </VerifonePayment_charge>
  </jobs>
</crontab>
```

checkIfCardsExpired method is checking card expiration. If Expire Date will be in 7 days, then email notification is sent to the customer. If card is expired profile is set as 'expired'