PayZen

e-payment gateway

SEPA Direct Debit

SDD Recurring

PayZen 2.5

Document version 2.5

Contents

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
2.5	Lyra Network	2/8/2016	- Additional information about the countries compatible with the SEPA payment method - Chapters added: - Managing chargeback - Manual cancellation of a recurring payment via the Back Office - Automatic cancellation of a recurring payment via Web Services
2.4	Lyra Network	11/23/2015	Chapter added: - **Setting up the notification to validate SEPA files sent to the acquirer**
2.3	Lyra Network	10/1/2015	Additional information: - Prerequisites for SEPA credit transfer. - Mandate signature by e-mail.
2.2	Lyra Network	9/18/2015	Chapters updated: - Using iframes - Registering a mandate - Creating one click payment - Saving a mandate and set a fixed subscription - Registering a mandate and making a payment - Updating a mandate and making a payment
2.1	Lyra Network	7/22/2015	- Shifting the payment gateway to version 2.4 Changes about the desired due date. Chapters updated: - Configuring delays for SEPA direct debit presentation - Capture delay - Saving a mandate and set a fixed subscription - Modifying the desired due date - Analyzing the payment outcome Chapters added: - Activating the pre-notification - Making one click payment (using an existing mandate) - Analyzing one click payment
2.0	Lyra Network	3/17/2015	- Shifting the payment gateway to version 2.3 - Refund by wire transfer - Receiving logs of outstanding payments by e-mail or FTP - Customization of the pre-notification e-mail for SEPA direct debit in the notification center
1.0	Lyra Network	9/16/2014	Initial version SEPA Direct Debit (SDD) - Recurring

Confidentiality

All information contained in this document is confidential. Its use outside the scope of this consultation or its disclosure to external parties is subject to preliminary approval by Lyra Network.

2. GETTING IN TOUCH WITH TECHNICAL SUPPORT

In case of a connection problem in the Back Office, go to "Forgotten password or locked account?".

For technical inquiries or support, you can reach us from Monday to Friday between 9am and 6pm:

by phone at:	0 811 708 709 from France, (*Azur Number – Cost of a local call from a landline*) +33 567 223 329 from abroad,
by e-mail:	*support@payzen.eu*

To facilitate the processing of your demands, you will be asked to communicate your shop ID (8 digits).

You will find this information in the subscription confirmation e-mail or in the Back Office (menu **Settings** > **Shop** > **Configuration**).

3. SEPA: PAYMENT BY DIRECT DEBIT

SEPA (Single Euro Payments Area), also called SEPA Direct Debit (SDD), is a payment-integration initiative of the European Union for simplification of payments denominated in euro (via credit transfers, direct debits and credit cards) for all European countries.

The creditor (merchant) proposes this payment method to the debtor (buyer) in order to automatically process recurring and one-off invoices.

SEPA Direct Debit is based on the SEPA Direct Debit Mandate.

Countries where SEPA payment method can be used:

Zone	Country
Eurozone EU countries	Germany, Austria, Belgium, Cyprus (Greek part), Spain, Estonia, Finland, France, Greece, Ireland, Italy, Luxembourg, Malta, Netherlands, Portugal, Slovakia, Slovenia, Croatia.
Non-Eurozone EU countries	Bulgaria, Denmark, Hungary, Latvia, Lithuania, Poland, Czech Republic, Romania, United Kingdom, Sweden.
EFTA (European Free Trade Association) countries	Iceland, Liechtenstein, Norway and Switzerland.

Table 1: Countries list

Note:

The French islands and the overseas departments (Guadeloupe, Martinique , Guiana, La Réunion and Mayotte as well as the territories of Saint Pierre and Miquelon, Saint-Barthélemy, the French part of St. Martin, the Principality of Monaco are part of SEPA zone.

On the other hand, the European microstates, Andorra, the Vatican City, San Marino and Gilbraltar are out of the SEPA zone.

3.1. Mandate information requirements

The SEPA Direct Debit mandate contains the following elements:

- The type of direct debits: one-off or recurring.
- The **UMR** (Unique Mandate Reference): a unique reference for each signed mandate.
- Debtor details (name).
- The **BIC** (Business Identifier Code) and the **IBAN** (International Bank Account Number) of the debtor: codes allowing to identify in a reliable way the beneficiary of a credit transfer or a direct debit.
- Creditor details (name and address).
- **Creditor ID** (SEPA creditor identifier): a unique code attributed to the company entitled to make direct debits. Creditor ID is issued by Banque de France (the French central bank) via a banking establishment.
- Mandate signature date.
- Signature of the debtor authorized by his bank to operate the account which is to be debited.

3.2. Distinguishing the SDD mandate type

There are two types of mandates:

- **One-off**

  The signed mandate applies to a single direct debit and will not result in a series of direct debits. It will not be possible to make other direct debits with the same mandate.

- **Recurring**

  The signed mandate applies to a series of direct debits. It becomes null after a period of 36 months with no payments.

The SDD mandate can be revoked by the debtor at any time.

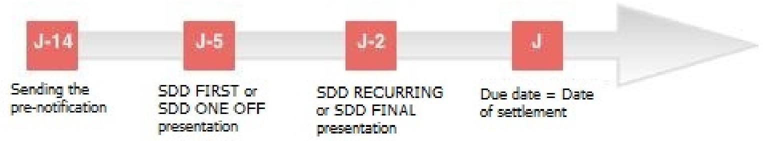
SEPA Direct Debit operates in "sequences". The following table lists the acceptable sequences for direct debits, each code corresponding to a mandate type.

Mandate type	Possible sequences for direct debit	Code
One-off	ONE-OFF: a unique direct debit not followed by other direct debits under the same mandate	OOFF
Recurring	FIRST: 1st direct debit of a series	FRST
	RECURRING: consecutive direct debits after the 1st of the series	RCUR
	FINAL: the last direct debit of the series	FNAL

Table 2: Possible sequences for direct debit

The delays for the presentation of SDD to the banks depend on the sequence type.

3.3. Identifying the delays for SEPA Direct Debit

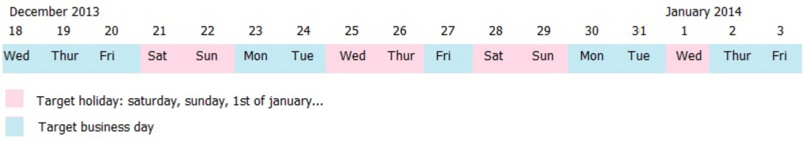

J-14
Sending the pre-notification
J-5
SDD FIRST or SDD ONE OFF presentation
J-2
SDD RECURRING or SDD FINAL presentation
J
Due date = Date of settlement

One of the prerequisites for the creditor for the emission of a SEPA direct debit is to notify the debtor about every SEPA direct debit at least 14 calendar days before the due date. However, this delay can be shortened. Upon a bilateral agreement, it can be reduced to:

- **5 working bank days** prior to the due date for a **one-off (OOFF) SEPA Direct Debit** or the first direct debit of a series (FRST).
- **2 working bank days** prior to the due date starting **from the second SEPA Direct Debit of a series** (Recurrent RCUR or Final FNAL).

Exemple of a SEPA Direct Debit submitted in December 2013

Taking holidays into account, December perfectly illustrates the delays that apply to payment collection via SEPA Direct Debit.



December 2013														January 2014		
18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3
Wed	Thur	Fri	Sat	Sun	Mon	Tue	Wed	Thur	Fri	Sat	Sun	Mon	Tue	Wed	Thur	Fri

🟥 Target holiday: saturday, sunday, 1st of january...

🟦 Target business day

A SEPA Direct Debit on January1st (due date) has to be presented at the debtor bank:

- On December 18th at the earliest, that is **14 calendar days** before the due date.
- On December 23rd at the latest, that is **5 bank business days** before the due date for a **one-off direct debit**.
- On December 30th at the latest, that is **2 bank business days** before the due date for a **recurring direct debit**.

January 1st is a holiday. This due date does not correspond to a bank business day. The SEPA direct debit order will be executed on the following bank business day, that is January 2nd.

3.4. Defining the steps of the SEPA direct debit

The interaction between the different parties involved into SEPA Direct Debit can be illustrated as follows:

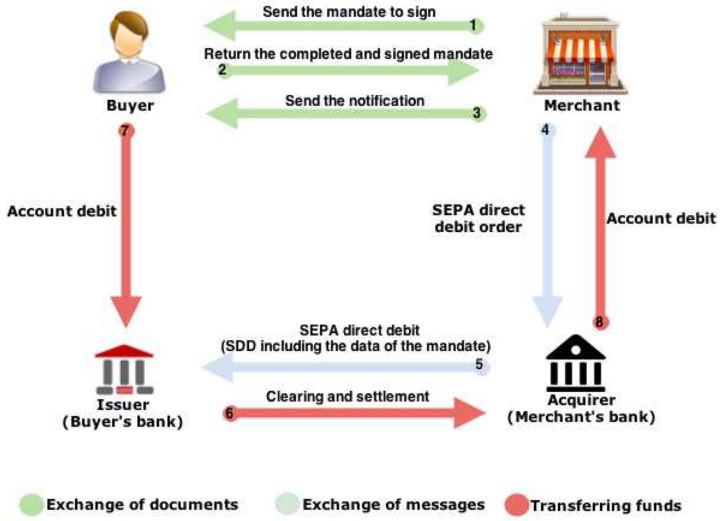


Figure 1: Overview of the data exchange process for SEPA direct debit

Stages 1, 2 and 3 – exchange of documents

The SDD is based on a SEPA mandate. Your debtor must sign this mandate in order to prove that he or she consents to the creditor debiting the debtor's account. Each mandate is identified by a **U**nique **M**andate **R**eference that allows to identify the contract to which the SEPA direct debits are attributable.

The creditor must pre-notify the debtor of the upcoming direct debit at least 14 calendar days before the due date.

Stages 4 and 5 – exchange of messages

The creditor must inform his or her bank about the direct debit order. The creditor's bank will then contact the debtor's bank to proceed with the direct debit.

Stages 6,7 and 8 – transferring funds

Funds from the debtor's bank account will be debited and transferred to the creditor's bank account.

3.5. Identifying the transmission channel of SEPA direct debit orders to the bank

SEPA Direct Debit uses the **EBICS protocol** (Electronic Banking Internet Communication Standard).

There are two types of EBICS protocols:

- **EBICS T with "detached signature"**: the creditor needs to connect to the bank account and manually validate direct debit files. This **communication protocol is used by the payment gateway**.
- **EBICS TS with "attached signature"**: prior to transferring direct debit orders to the bank, the creditor signs them with a certificate delivered by a certification authority.

Please contact your bank to know more about the terms of SEPA direct debit order transfers.

3.6. Managing chargeback

If a problem occurs after a direct debit, your bank will return a SEPA reason code.

Setting up chargeback flow

Bank side:

Activate the **SEPA Direct Debit chargeback flow** via your bank.

PayZen side:

Please ensure that your PayZen offer contains the Visual reconciliation of the chargebacks.

Its use is automatic. Once the **EBICS** connection and the **SEPA Direct Debit chargeback flow** activated, your payment gateway keeps you informed about chagebacks.

Viewing chargebacks in the Back Office

1. Connect to your Back Office. *https://secure.payzen.eu/vads-merchant/*
2. Go to the following menu: **Management** > **Transactions**.
3. Select **Captured transactions** tab.
   Chargebacks are visible from the **Captured transactions**tab.
4. Via the search tool, select the **Disputes** criteria.
   Disputes criteria is in the **Status group**.
5. Select **Yes** to filter results.
6. Click **Search**.
7. Double-click on a SEPA transaction to see its details.
   Details tab is displayed by default.
8. See in the **Transaction life cycle** group the **Chargeback reason**.
   A return code is associated to the chargeback.
   See the table below to analyze the SEPA Direct Debit Reason Codes and the action to perform.

SEPA Direct Debit Reason Codes table

ISO Code	Description	Action to perform
AC01	**Account identifier incorrect** - The IBAN or BIC code provided by the buyer is incorrect. - The account is not in euros.	Contact the buyer to obtain the correct IBAN. Via the Back Office, update the token. Resubmit the payment (see next chapter).
AC04	**Closed Account** The account has been closed. The buyer may have used an old account number or closed the account since setting up the mandate.	Contact the buyer to get new account details. Via the Back Office, update the token. Resubmit the payment (see next chapter).
AC06	**Account blocked** The account has been blocked for Direct Debits either by: - the bank - the buyer	Contact the buyer to ask for an alternative account/ payment method. Via the Back Office, update the token. Resubmit the payment (see next chapter).
AG01	**Direct debit forbidden on this account for regulatory reasons** A SEPA Direct Debit cannot be set up on this type of account. Eg.: savings account.	Contact the buyer for information about the account to be used. Via the Back Office, update the token. Resubmit the payment (see next chapter).
AG02	**Payment transaction code incorrect** Technical error	Contact the payment gateway.
AM04	**Insufficient funds** The buyer's bank could not pay the Direct Debit due to insufficient funds.	Contact the buyer to ensure they add funds to their account and resubmit the payment.
AM05	**Duplicate collection** The collection has already been made. This could either be due to an incorrect transaction ID or if two FRST transactions have been submitted.	Check whether the collection is really duplicated.
BE05	**Creditor Identifier incorrect** The Creditor Identifier provided is incorrect or is invalid.	Inform the correct ICS (Creditor Identifier) to the payment gateway (ADV).
FF01 (previous MD03 ISO code)	**Invalid file format** Technical error	Contact the payment gateway.
FF05	**Direct Debit type incorrect** Technical error	Contact the payment gateway.
MD01	**No valid mandate** The mandate is no longer live on the buyer's account. Mandate may have been cancelled by the buyer. Also used when a customer requests a refund, stating it was an unauthorized transaction. This can occur up to 13 months after settlement.	Check the expiry date in the Back Office. Contact the buyer for a a refund.
MD02	**Mandate data missing or incorrect** Technical issue.	Contact the payment gateway.
MD03	**Invalid file format** Technical issue.	Contact the payment gateway.
MD06	**Disputed authorized transaction** The buyer has claimed a refund for an authorized transaction. May be received up to 8 weeks after settlement.	Contact the buyer.
MD07	**End Customer Deceased** You have attempted to set up a mandate on the account of someone who is deceased. Extremely rare.	End the agreement with the deceased buyer.

ISO Code	Description	Action to perform
MS02	**Debtor refuses collection** The buyer refuses this particular collection. This code may be received pre- or post-settlement, depending on how quickly the buyer bank responds to the refusal.	Contact the buyer.
MS03	**Reason not specified** One of the most common error codes. Often used when banks choose not to issue more specific reason codes, such as MD07 and AM04, for data protection reasons.	Contact the buyer to request if he/she can contact his/her bank to know the reason.
RC01	**Invalid BIC** The BIC code provided by the buyer is incorrect.	Contact the buyer to obtain the correct BIC. Via the Back Office, update the token. Resubmit the payment (see next chapter).
RR01	**Regulatory requirements - Debtor account insufficient or missing** Technical issue.	Contact the payment gateway.
RR02	**Regulatory requirements - Debtor name missing** Technical issue.	Contact the payment gateway.
RR03	**Regulatory requirements - creditor name missing** Technical issue.	Contact the payment gateway.
RR04	**Regulatory reason** RR04 can only be used for a regulatory reason other than RR01, RR02 or RR03.	Contact your bank and request that they contact the buyer's bank to find out the rejection reason.
SL01	**Specific service offered by the debtor bank** The request clashes with specific instructions the buyer has put on it account. Eg.: the creditor is blacklisting.	Contact the buyer.
TM01	**File received after the limited time** Limited time passed.	Contact the payment gateway.

Table 3: ISO Code and SEPA Reason

Resubmit a payment

To resubmit a payment, perform a MOTO payment.

1. Connect to your Back Office. *https://secure.payzen.eu/vads-merchant/*

2. Go to the following menu: **Management > Transactions**.

3. Click the **MOTO payment** button located on the upper part of the transactions details view.

4. Select SEPA Direct Debit payment method.

5. Enter the token.

6. Fill in the necessary fields to complete the payment.
   Note:
   The next payments will automatically use the new bank data.

4. SELECTING A PAYMENT METHOD

The **UMR** (Unique Mandate Reference) is called:

- **Buyer ID** on the payment page,
- **Buyer ID** in the Back Office.

4.1. Prerequisites

To be able to use this payment method, the merchant must:

- Conclude a subscription **EBICS T** contract with his/her bank or ask for the creation of a new user on the existing EBICS connection.
- Subscribe to the **Payment by SEPA Direct Debit** option via PayZen.
- Inform PayZen about the Creditor ID **(ICS)**.

To receive the logs of outstanding payments by e-mail or FTP, the merchant must:

- Subscribe to receive the list of outstanding payments at his/her bank.
- Subscribe to the option **Bank reconciliation** via PayZen.

Prerequisites for SEPA credit transfer:

- Activate the SCT Service on your EBICS connection via your bank.

Storing and archiving signed mandates

The mandate signature is done by ticking **By checking this box, you agree that you have read and understood the details of this money order and that you append your signature**.

The mandate is saved on a local disk. The creditor (merchant) stores and archives the mandate.

Furthermore, it is possible, according the shop options, to enter a code received by SMS or e-mail

The mandate is legally archived in the electronic safe in the Deposit and Consignment Office.

In case of consignment, this signature serves you as guarantee that no changes have been made to the mandate by a third party. The Deposit and Consignment Office provides lines of evidence that can not be challenged or objected.

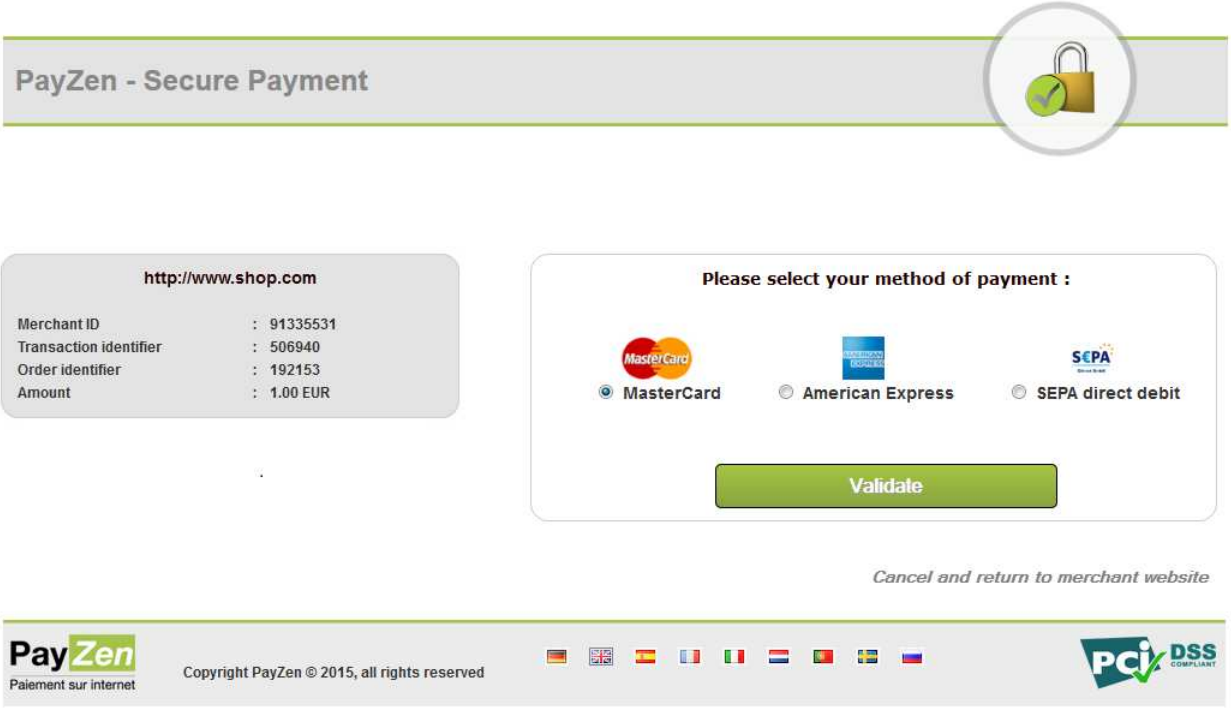
If you wish, you can deactivate the mandate signature by SMS or by e-mail. To do this, contact the customer service of the payment gateway.

4.2. Defining the steps of an automatic direct debit on the payment page

The recurring mandate applies to a series of direct debits.

SEPA Direct Debit can be illustrated as follows:

1. The debtor selects **SEPA Direct Debit**.



PayZen - Secure Payment

http://www.shop.com

Merchant ID	:	91335531
Transaction identifier	:	506940
Order identifier	:	192153
Amount	:	1.00 EUR

Please select your method of payment :

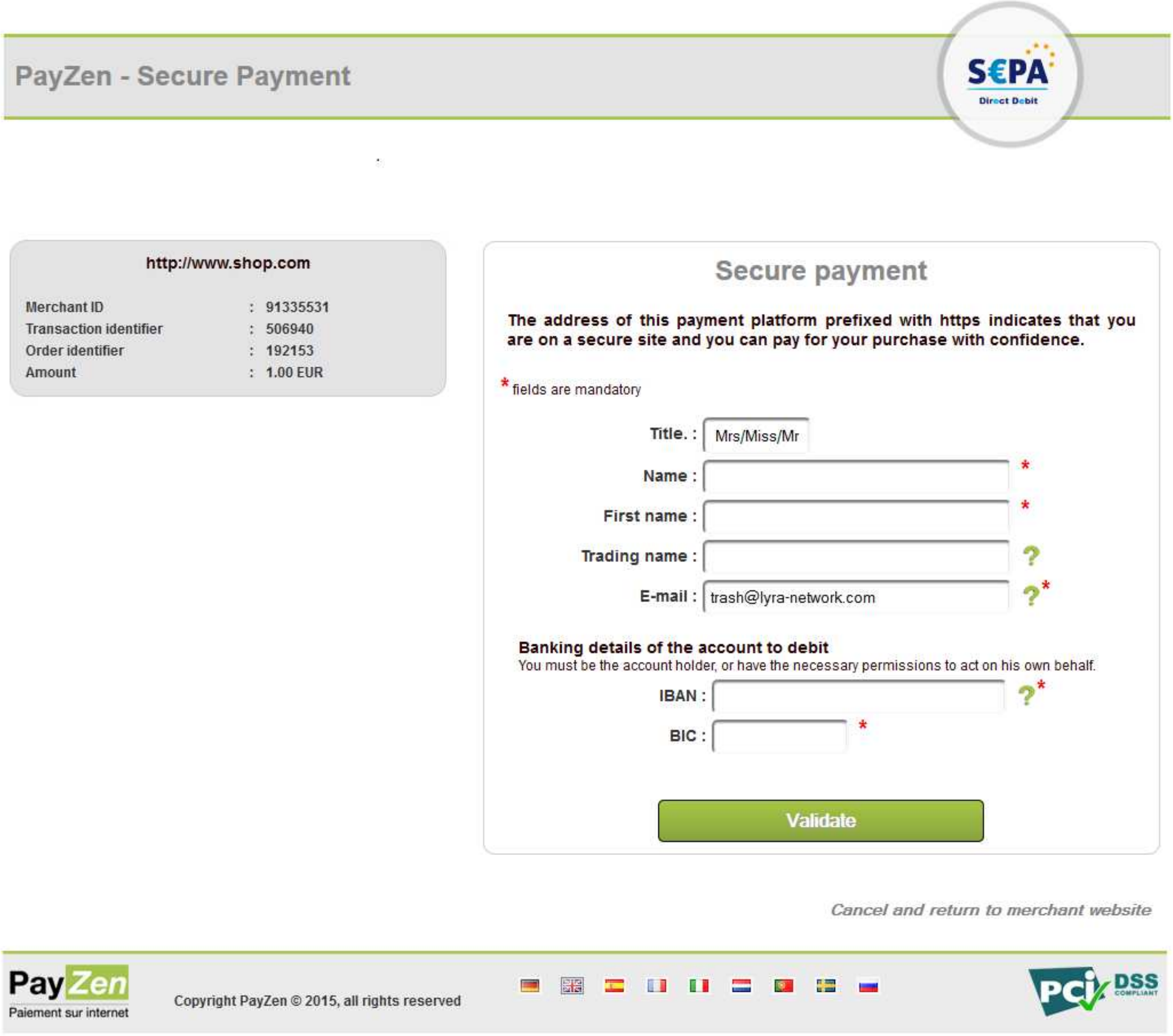
● MasterCard ○ American Express ○ SEPA direct debit

Validate
Cancel and return to merchant website
PayZen
Paiement sur internet
Copyright PayZen © 2015, all rights reserved
PCI DSS COMPLIANT

Figure 2: Selecting a payment method

2. The debtor clicks on **Validate**.

3. The next page displays the transaction summary (**Merchant ID** and **Amount**). The debtor must fill in the **First name**, **Last name**, **e-mail**, **IBAN**, **BIC** and **phone number** if this one is not pre-filled.



PayZen - Secure Payment

http://www.shop.com

Merchant ID	:	91335531
Transaction identifier	:	506940
Order identifier	:	192153
Amount	:	1.00 EUR

Secure payment

The address of this payment platform prefixed with https indicates that you are on a secure site and you can pay for your purchase with confidence.

* fields are mandatory

Title. : Mrs/Miss/Mr

Name : *

First name : *

Trading name : ?

E-mail : trash@lyra-network.com ?*

Banking details of the account to debit

You must be the account holder, or have the necessary permissions to act on his own behalf.

IBAN : ?*

BIC : *

Cancel and return to merchant website

Copyright PayZen © 2015, all rights reserved

Figure 3: Filling in the debtor's bank account details

Note:

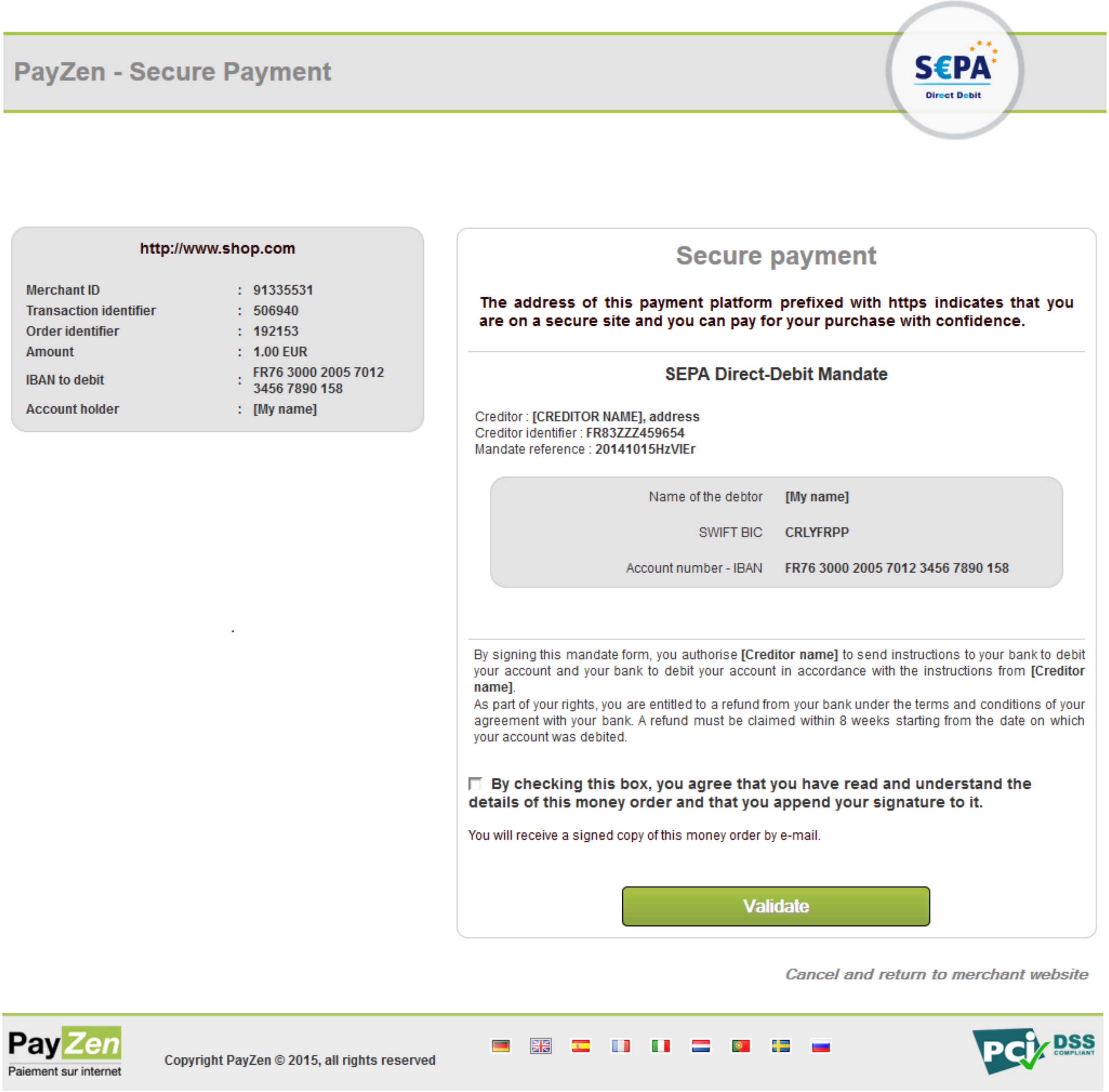
Another form of signature is available (signature by check box). For more information, please contact your payment gateway customer service.

4. Once the fields are filled in, the debtor clicks on **Validate**.

5. Several tests are made to verify the validity of the filled in IBAN and BIC.

   If the data is incorrect, the following message appears: **The specified bank account is not compatible with this payment method**.

   If the IBAN test is completed successfully, the SEPA direct debit mandate with all the required information is displayed. The debtor must verify this information.



PayZen - Secure Payment

http://www.shop.com

Merchant ID	:	91335531
Transaction identifier	:	506940
Order identifier	:	192153
Amount	:	1.00 EUR
IBAN to debit	:	FR76 3000 2005 7012 3456 7890 158
Account holder	:	[My name]

Secure payment

The address of this payment platform prefixed with https indicates that you are on a secure site and you can pay for your purchase with confidence.

SEPA Direct-Debit Mandate

Creditor : **[CREDITOR NAME], address**
Creditor identifier : **FR83ZZZ459654**
Mandate reference : **20141015HzVIEr**

Name of the debtor	**[My name]**
SWIFT BIC	**CRLYFRPP**
Account number - IBAN	**FR76 3000 2005 7012 3456 7890 158**

By signing this mandate form, you authorise **[Creditor name]** to send instructions to your bank to debit your account and your bank to debit your account in accordance with the instructions from **[Creditor name]**.
As part of your rights, you are entitled to a refund from your bank under the terms and conditions of your agreement with your bank. A refund must be claimed within 8 weeks starting from the date on which your account was debited.

□ **By checking this box, you agree that you have read and understand the details of this money order and that you append your signature to it.**

You will receive a signed copy of this money order by e-mail.

Validate

Cancel and return to merchant website

Copyright PayZen © 2015, all rights reserved

Figure 4: SMS Mandate validation example

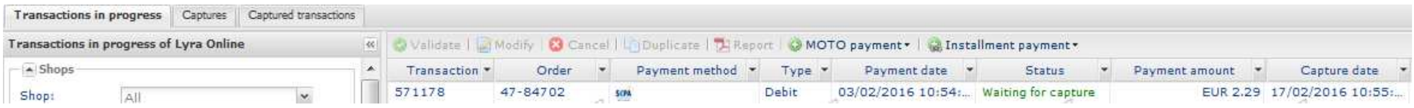
6. To validate this mandate, the debtor must:
   - enter the **Code** received by SMS (or e-mail depending on the shop option).
   - check **By checking this box, you agree that you have read and understood the details of this money order and that you append your signature to it**.
7. The debtor clicks on **Validate**.
8. The overview of the details will appear.

4.3. Viewing SEPA transactions in the Back Office

Transactions can be viewed in the Back Office accessible via **Management** > **Transactions**.

As long as the due date has not arrived, transactions can be viewed in the **Transactions in progress** tab.

Once the capture has been made, transactions appear in the **Captured transactions** tab.



Transactions in progress of Lyra Online

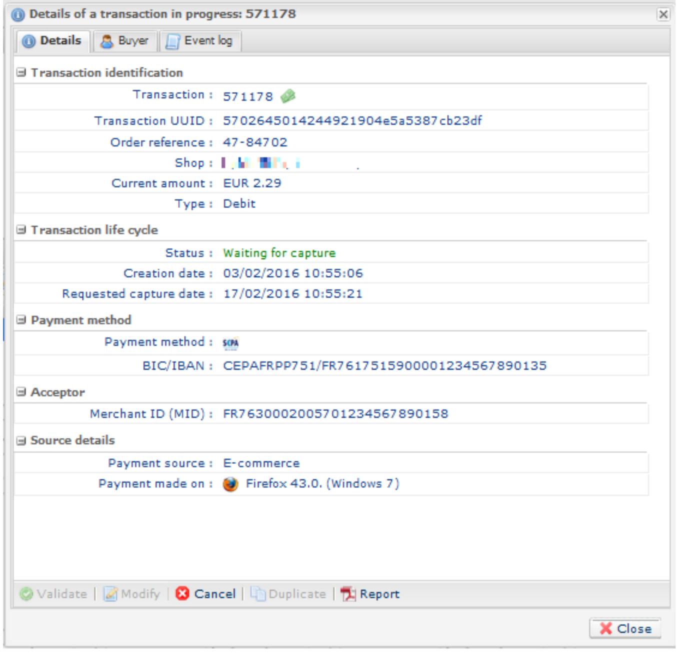
Shops

Shop: All

Transaction	Order	Payment method	Type	Payment date	Status	Payment amount	Capture date
571178	47-84702	SEPA	Debit	03/02/2016 10:54:...	Waiting for capture	EUR 2.29	17/02/2016 10:55:...

Figure 5: Transactions in progress tab

Double-click on a SEPA transaction to see its details.



Details of a transaction in progress: 571178

Transaction identification

Transaction : 571178

Transaction UUID : 57026450142449219O4e5a5387cb23df

Order reference : 47-84702

Shop : [redacted]

Current amount : EUR 2.29

Type : Debit

Transaction life cycle

Status : Waiting for capture

Creation date : 03/02/2016 10:55:06

Requested capture date : 17/02/2016 10:55:21

Payment method

Payment method : SEPA

BIC/IBAN : CEPAFRPP751/FR7617515900001234567890135

Acceptor

Merchant ID (MID) : FR7630002005701234567890158

Source details

Payment source : E-commerce

Payment made on : Firefox 43.0. (Windows 7)

Figure 6: Details of a SEPA transaction in progress

4.4. Viewing mandate details in the Back Office

The details of the completed and signed mandates are accessible via the Back Office.

The details can be viewed in the **Management > Recurring payments** menu **> Token** tab by selecting a **Token**.

You may also receive the mandate by e-mail (right-click on the token > **Receive the mandate by e-mail)**.

The mandate is translated only into the languages of the European Union: French, German, English, Spanish, Italian, Dutch, Polish, Portuguese and Swedish.

4.5. Notifying the debtor

Two e-mails can be sent to the debtor by the payment gateway:

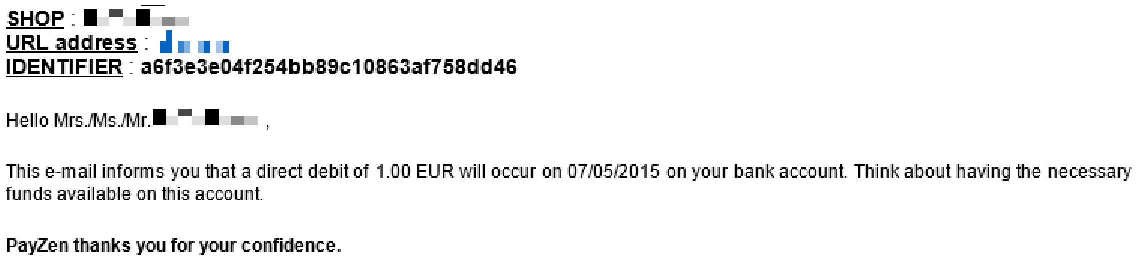
A pre-notification e-mail

By default, the pre-notification e-mail is deactivated (see chapter **Activating the pre-notification**).

The payment gateway can send an e-mail notification to the debtor maximum 14 business days before the due date.

The two parties may decide to reduce this delay (see chapter **Identifying the SEPA direct debit submission deadlines**).

Pre-notification e-mail template:



SHOP : [redacted]
URL address : [redacted]
IDENTIFIER : **a6f3e3e04f254bb89c10863af758dd46**

Hello Mrs./Ms./Mr. [redacted] ,

This e-mail informs you that a direct debit of 1.00 EUR will occur on 07/05/2015 on your bank account. Think about having the necessary funds available on this account.

PayZen thanks you for your confidence.

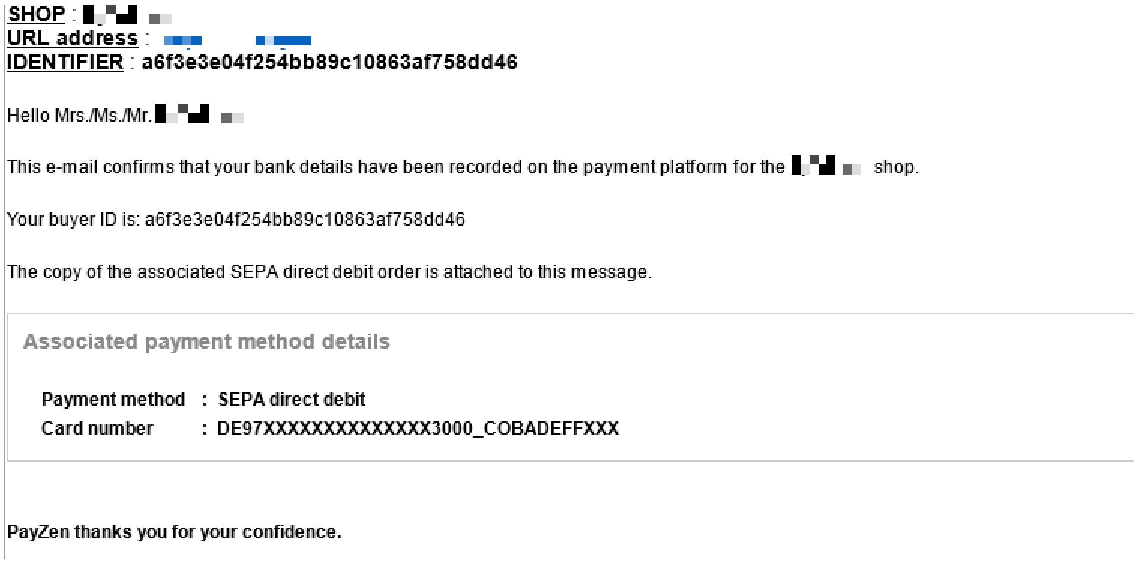
Note:

The pre-notification e-mail template can be customized (see chapter **Customizing the pre-notification e-mail**).

An e-mail to notify the payment

The payment gateway notifies the debtor by e-mail to confirm that his/her request for direct debit has been registered.

Example of notification e-mail:



SHOP : [redacted]
URL address : [redacted]
IDENTIFIER : **a6f3e3e04f254bb89c10863af758dd46**

Hello Mrs./Ms./Mr. [redacted]

This e-mail confirms that your bank details have been recorded on the payment platform for the [redacted] shop.

Your buyer ID is: a6f3e3e04f254bb89c10863af758dd46

The copy of the associated SEPA direct debit order is attached to this message.

Associated payment method details

Payment method	:	SEPA direct debit
Card number	:	DE97XXXXXXXXXXXXXX3000_COBADEFFXXX

PayZen thanks you for your confidence.

Activating the pre-notification

To activate the pre-notification e-mail:

1. Connect to: *https://secure.payzen.eu/vads-merchant/*.
2. Go to the following menu: **Settings** > **Notification rules**
3. Select the tab **E-mail sent to the buyer**.
4. Right-click on **Pre-notification e-mail of SEPA direct debit to the buyer**.
5. Select **Manage the rule**.

Customizing the pre-notification e-mail

The pre-notification e-mail template can be customized. To do this:

1. Connect to your Back Office (*https://secure.payzen.eu/vads-merchant/*).
2. Click on **Settings** > **Notification rules** > **E-mail sent to the buyer** tab.
3. Right-click on **SEPA direct debit pre-notification e-mail sent to the buyer**.
4. Select **Manage the rule** in the menu.
5. In **Buyer e-mail settings**, select the language of the e-mail and click on **Restore some default values** to customize the contents of the e-mail.

   Before saving your modifications, you can preview the pre-notification e-mail by clicking on the **Preview the e-mail** button situated in the bottom of the **Management of the rule** dialog box.
6. Click on **Save**.

4.6. Importing SEPA direct debit mandates

You can import SEPA direct debit mandates.

For more information, please contact the technical contact.

5. GENERATING A REQUEST FOR AN AUTOMATIC PAYMENT VIA DIRECT DEBIT

A request for an automatic payment via direct debit consists in registering a recurring mandate and defining the characteristics of a subscription (amounts, frequency, etc.).

Mandate registration is only made via a payment form.

To use a form you must:

- establish interaction with the payment gateway (see chapter **Establishing interaction with the payment gateway**),
- generate and send the completed form (see chapters **Generating a payment form** and **Sending the payment request**),
- analyze the response (see chapter **Analyzing the payment result**).

To define the subscription, depending on your needs you may choose one of the following options:

Subscription with fixed amounts and due dates.

In this case, the fields of the mandate registration form will be filled in with the values defined in the subscription.

Subscription with flexible amounts and/or due dates.

In this case, once the mandate form is registered, use a file sharing service to proceed to direct debits.

The file transfer process consists of four steps:

- generate the file
- upload the file
- download the return file
- analyze the return file

6. ESTABLISHING INTERACTION WITH THE PAYMENT GATEWAY

The merchant website and the payment gateway interact by exchanging data.

To create a payment, this data is sent in an HTML form via the buyer's browser.

At the end of payment, the result can be transmitted to the merchant website in two ways:

- automatically by means of a notification called Instant Notification URL (also called IPN), see chapter **Setting up notifications**,
- via the browser when the buyer clicks on the button to return to the merchant website, .

To guarantee the security of the exchange, the data is signed with a certificate only known to the merchant and the payment gateway.

6.1. Setting up the payment page URL

The merchant website interacts with the payment gateway by redirecting the buyer to the following page:

https://secure.payzen.eu/vads-payment/

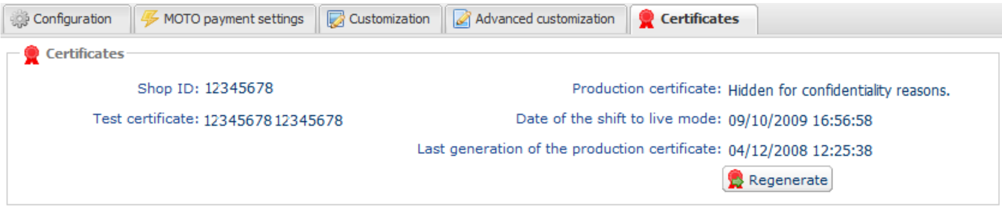
6.2. Identifying yourself during data exchange

To be able to interact with the payment gateway, the merchant needs to have:

- **The shop ID**: allows to identify the merchant website during data exchange. Its value is transmitted in the **vads_site_id** field.
- **The certificate**: allows to compute the numerical signature transmitted in the **signature** field.

To retrieve these values:

1. Connect to your Back Office: *https://secure.payzen.eu/vads-merchant/*
2. Click on **Settings > Shop**.
3. Select the **Certificates** tab.



Shop ID: 12345678

Test certificate: 12345678 12345678

Production certificate: Hidden for confidentiality reasons.

Date of the shift to live mode: 09/10/2009 16:56:58

Last generation of the production certificate: 04/12/2008 12:25:38

Figure 7: View the shop ID and the certificate.

Two types of certificates are available:

- The **test certificate** for generating the form signature in test mode.
- The **production certificate** for generating the form signature in production mode.

6.3. Choosing between Test and Production mode

The choice between **TEST** or **PRODUCTION** mode can be made in the **vads_ctx_mode** field (see chapter **Generating a payment form**).

- The **TEST** mode allows to make test payments.

  It is available at all times, even after a production certificate has been generated.

  If you create a new merchant website (or have access to the acceptance testing environment), you can make tests without affecting the website that is currently in production.

  The TEST transactions can be viewed in the Back Office accessible via **Management** > **TEST Transactions.**

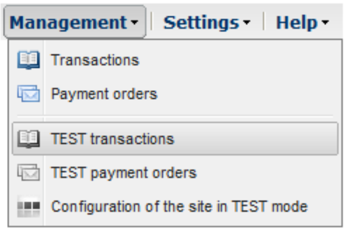


Figure 8: Viewing test transactions

- The **PRODUCTION** mode is only available once the production certificate has been generated (see chapter **Activating the shop in PRODUCTION mode**).

  It allows to make real payments.

  The PRODUCTION transactions can be viewed in the Back Office accessible via **Management** > **Transactions**.

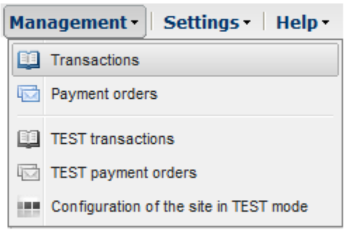


Figure 9: Viewing production transactions

6.4. Managing the interaction with the merchant website

Two types of URLs are used to manage the interaction with the merchant website:

- **Instant Payment Notification (IPN) URL**,
- **Return URL** to the merchant website.

Instant Payment Notification (IPN) URL

The payment gateway automatically informs the merchant website about the payment result. The data is sent in **POST** mode.

The payment gateway is able to contact the merchant website regardless of the protocol (http or https) that was used.

To process these notifications, the merchant must **create a page** on his/her website that:

- analyzes the data received in **POST** mode,
- checks that the received information is complete by computing the signature,
- checks that the notification is not a duplicate (e.g. notification returned from the Back Office),
- triggers an update of its database (order status, stock status, etc.),
- sends e-mails to the buyer (invoice, order tracking, etc.).

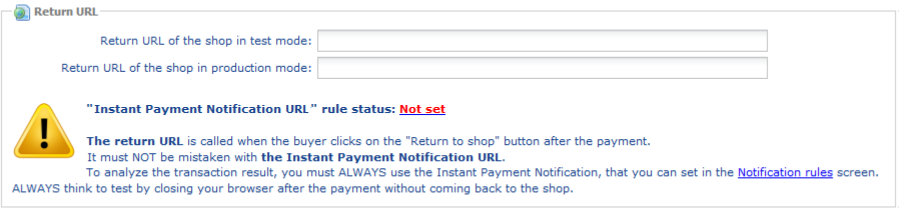
The processing time directly impacts the delay for displaying the payment summary. The longer the processing takes, the later the summary will be shown.

To receive notifications, the merchant must **set up** the notification rules in his/her Back Office (see chapter **Setting up notifications**).

In case of an issue during the interaction with the merchant website, the payment gateway sends an e-mail to the shop administrator stating the reason of the error (HTTP error, etc.) and the instructions for returning the notification from the Back Office.

Return URL to the merchant website

The merchant can set the "default" return URLs in the Back Office accessible via **Settings** > **Shop** > **Configuration** tab):



Return URL

Return URL of the shop in test mode:

Return URL of the shop in production mode:

"Instant Payment Notification URL" rule status: Not set

The return URL is called when the buyer clicks on the "Return to shop" button after the payment.
It must NOT be mistaken with **the Instant Payment Notification URL.**
To analyze the transaction result, you must ALWAYS use the Instant Payment Notification, that you can set in the Notification rules screen.
ALWAYS think to test by closing your browser after the payment without coming back to the shop.

Figure 10: Setting up return URLs

The merchant can set up a different return URL for each mode.

By default, the buyer is redirected to the return URL regardless of the payment outcome.

If no URL has been set up, the main URL of the shop will be used for redirection (**URL** parameter defined in the **Details** section of the shop).

The merchant will be able to override this setting in his/her payment form (see chapter **Setting up redirection URLs**).

Note:

The status of the "Instant Notification URL" (IPN) rule appears on this screen. If the URL has not been set up, make sure to configure it (see chapter **Setting up notifications**).

6.5. Managing security

There are several ways to guarantee the security of online payments.

Managing security by means of a signature

The integrity of shared information is preserved by the exchange of numerical signatures between the payment gateway and the merchant website.

The payment gateway and the merchant website interact via HTML forms.

A form contains a list of specific fields (see **Generating a payment form**) used for computing a numerical signature (see chapter **Computing the signature**).

The result of this computation must be sent in the **signature** field.

We use the "SHA-1" hash function for signature encryption.

SHA-1 is available in most languages used for web application development.

Depending on the language, you might need to code the SHA-1 hash functions yourself (e.g. as in ASP).

Here is an example for the result of SHA-1 for the character "a":

SHA1(" a ") = 86f7e437faa5a7fce15d1ddcb9eaeaea377667b8

Modeling security mechanisms:

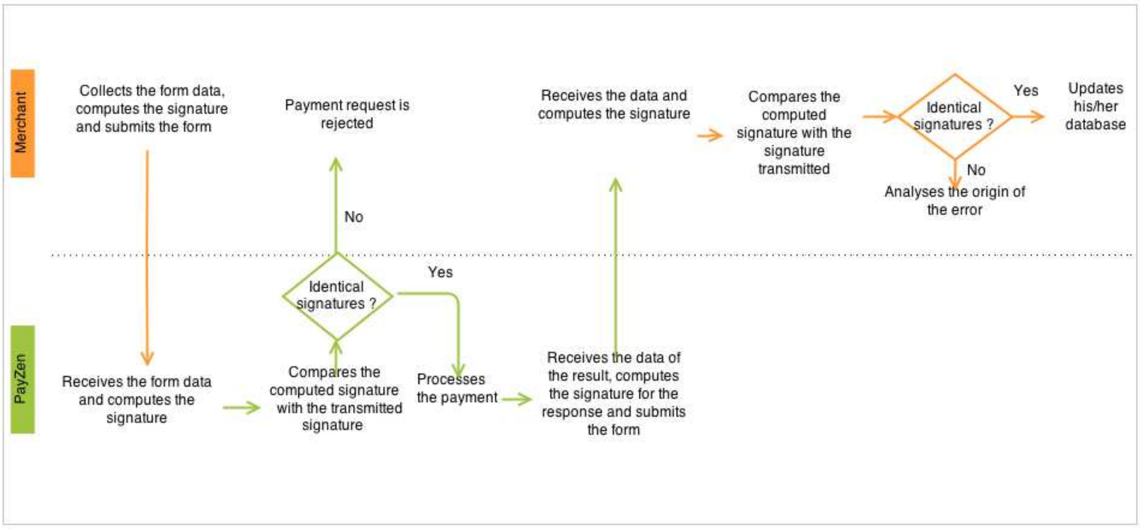

Merchant
Collects the form data, computes the signature and submits the form
Payment request is rejected
Receives the data and computes the signature
Compares the computed signature with the signature transmitted
Identical signatures ?
Yes
Updates his/her database
No
Analyses the origin of the error
No
Yes
Identical signatures ?
PayZen
Receives the form data and computes the signature
Compares the computed signature with the transmitted signature
Processes the payment
Receives the data of the result, computes the signature for the response and submits the form

Figure 11: Diagram of a security mechanism

1. The merchant website collects the form data and computes the signature.
2. The merchant website submits the form to the payment gateway.
3. The payment gateway receives the form data and computes the signature.
4. The payment gateway compares the computed signature with the signature transmitted by the merchant website.

5. If the signatures are different, the payment request is declined.

   If they are not, the payment gateway processes the payment.
6. The payment gateway receives the data of the result and computes the signature for the response.
7. Depending on the shop configuration (see chapter **Setting up notifications**), the payment gateway submits the payment result to the merchant website.
8. The merchant website receives the data and computes the signature. It compares the computed signature with the signature transmitted by the payment gateway.
9. If the signatures are different, the merchant analyses the origin of the error (error in computation, attempted fraud, etc).

   If they are not, the merchant proceeds to update his/her database (stock status, order status, etc).

Storing the production certificate

Starting with the first payment with a real credit card, the production certificate is masked for security reasons.

We strongly recommend you to store this certificate in a secure place (encrypted file, database, etc).

If the certificate is lost, the merchant will be able to generate a new one in the Back Office.

Remember that the production certificate can be viewed in the Back Office (**Settings** > **Shop** > **Certificates** tab.

Managing sensitive data

Online payments are regulated by strict rules (PCI-DSS certification).

As a merchant, you must make sure to never disclose data that might resemble a credit card number. Your form will be rejected (code 999 - Sensitive data detected).

Particularly avoid using order numbers between 13 and 16 digits long and numbers starting with 3, 4 or 5.

6.6. Using iframes

An iframe (in-line frame) is used to show one web page within another web page.

Examples of code:

```
<FRAMESET>
 <FRAME SRC="https://secure.payzen.eu/vads-payment/">
</FRAMESET>
```

```
<iframe src="https://secure.payzen.eu/vads-payment/"></iframe>
```

6.7. Setting up notifications

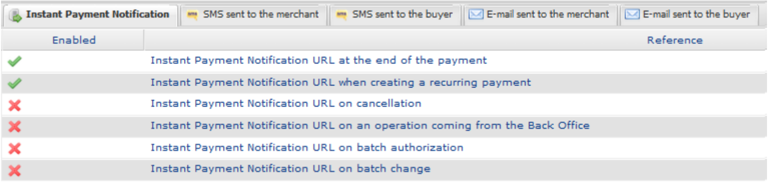
The Back Office allows to manage the events that will trigger a call for the merchant website and to configure the URL of the page to contact.

Setting up notifications

Several types of notifications are provided in the Back Office. They allow the configuration of the URL of the page to contact and the management of the events (payment abandoned by the buyer, payment canceled by the merchant, payment validated by the merchant, etc.) that will trigger a call to the merchant website.

To access notification rule management:

1. Connect to: *https://secure.payzen.eu/vads-merchant/*.
2. Go to the following menu: **Settings** > **Notification rules**.



Enabled	Reference
✓	Instant Payment Notification URL at the end of the payment
✓	Instant Payment Notification URL when creating a recurring payment
✗	Instant Payment Notification URL on cancellation
✗	Instant Payment Notification URL on an operation coming from the Back Office
✗	Instant Payment Notification URL on batch authorization
✗	Instant Payment Notification URL on batch change

Figure 12: Notification rules

Setting up the end of payment notification

To set up this notification:

1. Right-click on **Instant Payment Notification URL at the end of the payment**.

2. Select **Manage the rule**.

3. Enter the URL of your page into **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode.**

4. Enter the **E-mail address(es) to notify in case of failure**.

5. Set up the parameters for **Automatic retry in case of failure**.

   This option allows to automatically send notifications to the merchant website in case of failure, up to 4 times.

   For more information, see chapter **Activating the automatic retry** of the Payment Form Implementation Guide available here:

   https://payzen.io/

6. Save the modifications.

If the gateway is unable to access the URL of your page, an e-mail will be sent to the address specified in step 6 .

It contains:

- The HTTP code of the encountered error
- Parts of error analysis
- Its consequences
- Instructions to resend the notification to the URL specified in step 5 from the Back Office .

Setting up notifications in case of abandoned/canceled payments

It is possible to configure the settings so that the payment gateway systematically notifies the merchant website:

- When the buyer abandons/cancels a payment - via the **Cancel an return to shop** button.
- When the buyer has not completed the payment process and the payment session has expired.

  The maximum length of a payment session is 10 minutes.

To set up this notification:

1. Right-click on **Instant Payment Notification URL on cancellation**.
2. Select **Manage the rule**.
3. Enter the URL of your page into **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode**
4. Fill in the **E-mail address(es) to notify in case of failure**.
5. To specify several e-mail addresses, separate them by a semi-colon.
6. Set up the parameters for **Automatic retry in case of failure**.

   This option allows to automatically send notifications to the merchant website in case of failure, up to 4 times.

   For more information, see chapter **Activating the automatic retry** of the Payment Form Integration Guide available here :*https://payzen.io/*
7. Save the modifications.

If the payment gateway is unable to access the URL of your page, an e-mail will be sent to the address specified in step 6.

It contains:

- the HTTP code of the encountered error,
- analysis parts for the error,
- Instructions to resend the notification to the URL specified in step 5 from the Back Office .

Setting up notifications of the final outcome of a deferred payment

This notification is required to communicate the result of each due date.

To set up this notification:

1. Connect to: *https://secure.payzen.eu/vads-merchant/*.
2. Go to the following menu: **Settings** > **Notification rules** >[shop name].
3. Right-click on **Instant payment notification when creating a recurring payment**.
4. Select **Manage the rule**.
5. Enter the URL of your page into the fields **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode**.
6. Fill in the **E-mail address(es) to notify in case of failure**.

   In case of notification failure, e-mail are sent to this address.
7. To specify several e-mail addresses, separate them by a semi-colon.
8. Set up the parameters for **Automatic retry in case of failure**.

   This option allows to automatically send notifications to the merchant website in case of failure.

   Call attempts are programmed at fixed hours every 15 minutes (00, 15, 30, 45).

   After each failed attempt, a notification e-mail is sent to the e-mail address entered earlier.
9. Save the modifications.

If the payment gateway is unable to access the URL of your page, an e-mail will be sent to the shop administrator.

It contains:

- the HTTP code of the encountered error,
- parts of error analysis,
- instructions to resend the notification to the URL specified in step 5 from the Back Office.

Setting up the notification to validate SEPA files sent to the bank

The payment gateway can notify the merchant when a SEPA Direct Debit/SEPA Credit Transfer file order is sent to its bank.

This notification is required to inform the merchant that he must validate this transfer file with its bank in order to receive money.

This notification can be sent by **SMS** and/or **E-mail.**

- Setting up the SMS notification

1. Select **SMS sent to the merchant** tab.
2. Right click on **Delivery SMS confirmation - sending a Direct debit/credit transfer file**
3. Select **Manage the rule**.
   The **Management of the rule** dialog box appears.
4. Enter the phone number to notify in the **Settings** tab.
5. Click on **Save**.

- Setting up the e-mail notification

1. Select the tab **E-mail sent to the merchant**.
2. Right click on **Sending confirmation e-mail of a Direct debit/credit transfer file**.
3. Select **Manage the rule**.
   The **Management of the rule** dialog box appears.
4. Enter the e-mail address to notify in the **Setting** tab.
5. Click on **Save**.

Note:

Labels and messages can be customized.

Variable names are provided for information purposes. They are case sensitive.

6.8. Configuring delays for SEPA direct debit presentation

You can modify:

- The delay between the time of submission of the form and the desired due date (capture delay),
- The delay for sending a pre-notification to the debtor (buyer).

Capture delay

Can be modified in the Back Office (**Settings** > **Shop** > **Configuration**).

By default, the capture delay is set to 0 days.

It can be modified via the payment form (see chapter **Modifying the date of a direct debit**).

Any changes to the capture delay will cause the modification of the pre-notification date.

Pre-notification delay

Can be modified on request only via customer service.

By default, it is set to 14 calendar days.

Its value can be between 14 and:

- 6 days minimum for a one-off direct debit or for the first installment,
- 3 days minimum for a recurring direct debit.

7. GENERATING A PAYMENT FORM

To generate a payment request, you must create an HTML form as follows:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
 <input type="hidden" name="parameter1" value="value1" />
 <input type="hidden" name="parameter2" value="value2" />
 <input type="hidden" name="parameter3" value="value3" />
 <input type="hidden" name="signature" value="signature"/>
 <input type="submit" name="pay" value="Pay"/>
</form>
```

It contains:

The following technical elements:

- The `<form>` and `</form>` tags that allow to create an HTML form.
- The `method="POST"` attribute that defines the method used for sending data.
- The `action="https://secure.payzen.eu/vads-payment/"` attribute that defines where to send the form data.

Form data:

- Shop ID,
- Information about the payment depending on the use case ,
- Additional information, depending on your requirements ,
- Signature that certifies the integrity of the form.

This data is added to the form by using the `<input>` tag:

```
<input type="hidden" name="parameter1" value="value1" />
```

For setting the `name` and `value` attributes, see chapter **Data dictionary**.

All the data in the form must be encoded in **UTF-8.**

Special characters (accents, punctuation marks, etc.) will then be correctly interpreted by the payment gateway. Otherwise, the signature will not be computed correctly and the form will be rejected.

The **Pay** button that will allow to send data:

```
<input type="submit" name="pay" value="Pay"/>
```

7.1. Registering a mandate

The registration of a recurring mandate corresponds to creating a token.

The token can be later used for:

- performing a payment,
- creating a subscription,
- using a file sharing service. Using this form together with a file sharing service allows for each direct debit to have a different amount and due date.
- etc

In order to register a recurring mandate:

1. Use all the fields of the table below to create your payment form.

Field name	Description	Value
vads_site_id	Shop ID	Eg.: 12345678
vads_ctx_mode	Operating mode	**TEST** or **PRODUCTION**
vads_trans_date	Date and time of the payment form in UTC format	Eg.: 20140919130128
vads_action_mode	Acquisition mode for bank data.	**INTERACTIVE**
vads_page_action	Action required to **register** a mandate and perform a payment	**REGISTER**
vads_version	Version of the exchange protocol	**V2**
vads_cust_email	Buyer's e-mail address. **Note**: The buyer cannot modify the pre-filled e-mail address on the payment page. If the e-mail address in unknown, the buyer will have to enter it on the payment page.	Eg.: name@example.com

Table 4: Field list - Recurring payment by SEPA Direct Debit

2. Set **vads_page_action** to **REGISTER**.

3. Set **vads_action_mode** to **INTERACTIVE**.
   SEPA Direct Debit is not compatible with the SILENT mode.

4. If you would like to pre-fill the form, add the optional fields of the table below to create your payment form:

Field name	Description	Value
vads_cust_title	Allows to specify the buyer's title.	Eg.: **Mr** **Mrs** **Ms**
vads_cust_last_name	Allows to specify the buyer's last name.	Eg.: Dupont
vads_cust_first_name	Allows to specify the buyer's first name.	Eg.: Jean
vads_cust_cell_phone	Buyer cell phone number to receive the security code (by SMS). **Note**: The buyer cannot modify the pre-filled cell phone number on the payment page. Only GSM-users having recorded this phone number at the moment of subscription can receive an SMS with this access code for single use. If the phone number in unknown, the buyer will have to enter it on the payment page.	Eg.: 0612345678

Field name	Description	Value
vads_cust_country	Allows to specify the buyer's country code in compliance with the ISO 3166 standard.	Eg.: **FR**

Table 5: Field list - Recurring payment by SEPA Direct Debit

5. Set **vads_payment_cards** if you would like to pre-select the payment method on the merchant website.

   Note:

   - *If this list contains only one card type, the data entry page for this payment method will directly appear.*

     *Expected value for SEPA Direct Debit is **SDD**.*

   - *If there are more than one card types, the page with payment method selection will appear.*
   - *If this parameter is empty (recommended), the available payment methods (currencies, technical constraints, etc.) associated with the shop will be proposed.*

6. Add optional fields according to your requirements (see chapter **Using additional functions**).

7. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of a form:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_cust_email" value="name@example.com" />
<input type="hidden" name="vads_page_action" value="REGISTER" />
<input type="hidden" name="vads_site_id" value="70258842" />
<input type="hidden" name="vads_trans_date" value="20140919130128" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="54b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

7.2. Making one click Payment (using an existing mandate)

An existing and valid mandate is required. You must have registered a mandate (vads_page_action = REGISTER).

If you do not have one, see chapter **Registering a mandate**.

To make one click payment:

1. Use all the fields of the table below to create your payment form.

Field name	Description	Value
vads_site_id	Shop ID	Eg.: 12345678
vads_ctx_mode	Operating mode	**TEST** or **PRODUCTION**
vads_trans_date	Date and time of the payment form in UTC format	Eg.: 20150721130128
vads_action_mode	Acquisition mode for bank data.	**INTERACTIVE**
vads_page_action	Action required to **register** a mandate and perform a payment	**PAYMENT**
vads_identifier	Reminder of the payment token to debit	E.g.: 22e75ce992924f069279f42afe18da10
vads_version	Version of the exchange protocol	**V2**
vads_cust_email	Buyer's e-mail address. **Note**: The buyer cannot modify the pre-filled e-mail address on the payment page. If the e-mail address in unknown, the buyer will have to enter it on the payment page.	E.g.: name@example.com
vads_amount	Payment amount (in the smallest currency unit)	E.g.: 2990 for 29.90€
vads_currency	Code of the payment currency	978 for Euro

Table 6: Field list - Recurring payment by SEPA Direct Debit

2. Set **vads_page_action** to **PAYMENT**.

3. Populate the **vads_identifier** field with the mandate reference (UMR) that you obtained when registering the mandate.

4. Set **vads_action_mode** to **INTERACTIVE**.

   SEPA Direct Debit is not compatible with the SILENT mode.

5. Populate the **vads_currency** field with **978** (Euro code currency).

   Only the Euro currency is accepted for this payment method.

6. Populate the **vads_amount** field with an amount in the smallest currency unit (in cents for Euro).

7. If you would like to pre-fill the form, add the optional fields of the table below to create your payment form:

Field name	Description	Value
vads_cust_title	Allows to specify the buyer's title.	Eg.: • **Mr** • **Mrs** • **Ms**
vads_cust_last_name	Allows to specify the buyer's last name.	Eg.: Dupont
vads_cust_first_name	Allows to specify the buyer's first name.	Eg.: Jean
vads_cust_cell_phone	Buyer cell phone number to receive the security code (by SMS). **Note**: The buyer cannot modify the pre-filled cell phone number on the payment page. Only GSM-users having recorded this phone number at the moment of subscription can receive an SMS with this access code for single use. If the phone number in unknown, the buyer will have to enter it on the payment page.	Eg.: 0612345678
vads_cust_country	Allows to specify the buyer's country code in compliance with the ISO 3166 standard.	Eg.: **FR**

Table 7: Field list - Recurring payment by SEPA Direct Debit

8. Set **vads_payment_cards** if you would like to pre-select the payment method on the merchant website.

   Note:

   - *If this list contains only one card type, the data entry page for this payment method will directly appear.*

     *Expected value for SEPA Direct Debit is **SDD**.*

   - *If there are more than one card types, the page with payment method selection will appear.*
   - *If this parameter is empty (recommended), the available payment methods (currencies, technical constraints, etc.) associated with the shop will be proposed.*

9. Add optional fields according to your requirements (see chapter **Using additional functions**).

10. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

7.3. Saving a mandate and set a fixed subscription

To save a recurring mandate and create a subscription with fixed amounts and fixed dates:

1. Use all the fields in the table below to create your payment form.

Field name	Description	Value
vads_site_id	Shop ID	Eg.: 12345678
vads_ctx_mode	Operating mode	**TEST** or **PRODUCTION**
vads_trans_date	Date and time of the payment form in UTC format	Eg.: 20140919130128
vads_action_mode	Acquisition mode for bank data.	**INTERACTIVE**
vads_page_action	Action to perform	**REGISTER_SUBSCRIBE**
vads_version	Version of the exchange protocol	**V2**
vads_cust_email	Buyer's e-mail address. **Note**: The buyer cannot modify the pre-filled e-mail address on the payment page. If the e-mail address in unknown, the buyer will have to enter it on the payment page.	Eg.: name@example.com
vads_sub_effect_date	Subscription start date.	Eg.: 20141002
vads_sub_amount	Installment amount (in the smallest currency unit)	Eg.: 2990 for 29.90€
vads_sub_currency	Numerical code of the currency to be used	978 for Euro

vads_sub_desc	Rule for recurring payments to apply	Eg.: RRULE:FREQ=MONTHLY;
		BYMONTHDAY=10

Table 8: Field list - Recurring payment by SEPA Direct Debit

2. Set **vads_page_action** to **REGISTER_SUBSCRIBE**.

3. Set **vads_action_mode** to **INTERACTIVE**.

   SEPA Direct Debit is not compatible with the SILENT mode.

4. Set **vads_sub_effect_date**.

   For technical reasons, the due date must be at least **14 calandar days** from the submission date of the form (vads_trans_date).

5. Set **vads_currency** to **978** (Euro code currency).

   Euro is the only currency accepted.

6. Set **vads_sub_desc**.

   Examples of subscription rules:

   - To program installment payments taking place on the last day of each month for 12 months, the rule is:

   RRULE:FREQ=MONTHLY;COUNT=12;BYMONTHDAY=1

   - To program installment payments every three months until December 31th 2016, the rule is:

   RRULE:FREQ=YEARLY;BYMONTHDAY=1;BYMONTH=1,4,7,10;UNTIL=20161231

   - To program installment payments for **the last day of each month** for 12 months, the rule is:

   RRULE:FREQ=MONTHLY;BYMONTHDAY=28,29,30,31;BYSETPOS=-1;COUNT=12

7. If you would like to pre-fill the form, add the optional fields of the table below to create your payment form:

Field name	Description	Value
vads_cust_title	Allows to specify the buyer's title.	Eg.: • **Mr** • **Mrs** • **Ms**
vads_cust_last_name	Allows to specify the buyer's last name.	Eg.: Dupont
vads_cust_first_name	Allows to specify the buyer's first name.	Eg.: Jean
vads_cust_cell_phone	Buyer cell phone number to receive the security code (by SMS). **Note**: The buyer cannot modify the pre-filled cell phone number on the payment page. Only GSM-users having recorded this phone number at the moment of subscription can receive an SMS with this access code for single use. If the phone number in unknown, the buyer will have to enter it on the payment page.	Eg.: 0612345678
vads_cust_country	Allows to specify the buyer's country code in compliance with the ISO 3166 standard.	Eg.: **FR**

Table 9: Field list - Recurring payment by SEPA Direct Debit

8. Set **vads_payment_cards** if you would like to pre-select the payment method on the merchant website.

   Note:

   - *If this list contains only one card type, the data entry page for this payment method will directly appear.*

     *Expected value for SEPA Direct Debit is **SDD**.*

   - *If there are more than one card types, the page with payment method selection will appear.*
   - *If this parameter is empty (recommended), the available payment methods (currencies, technical constraints, etc.) associated with the shop will be proposed.*

9. Add optional fields according to your requirements (see chapter **Using additional functions**).

10. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of a form:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_cust_email" value="firstname.lastname@example.com" />
<input type="hidden" name="vads_page_action" value="REGISTER_SUBSCRIBE" />
<input type="hidden" name="vads_site_id" value="70258842" />
<input type="hidden" name="vads_sub_amount" value="2990" />
<input type="hidden" name="vads_sub_currency" value="978" />
<input type="hidden" name="vads_sub_desc" value="RRULE:FREQ=MONTHLY;BYMONTHDAY=1" />
<input type="hidden" name="vads_sub_effect_date" value="20141002" />
<input type="hidden" name="vads_trans_date" value="20140919130128" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="86b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

7.4. Registering a mandate and making a payment

In order to register a recurring mandate and make a payment:

1. Use all the fields in the table below to create your payment form.

Field name	Description	Value
vads_site_id	Shop ID	Eg.: 12345678
vads_ctx_mode	Operating mode	**TEST** or **PRODUCTION**
vads_trans_date	Date and time of the payment form in UTC format	Eg.: 20140919130128
vads_action_mode	Acquisition mode for bank data.	**INTERACTIVE**
vads_page_action	Action required to **register** a mandate and perform a payment	**REGISTER_PAY**
vads_version	Version of the exchange protocol	**V2**
vads_cust_email	Buyer's e-mail address. **Note**: The buyer cannot modify the pre-filled e-mail address on the payment page. If the e-mail address in unknown, the buyer will have to enter it on the payment page.	Eg.: name@example.com
vads_amount	Payment amount (in the smallest currency unit)	Example: 2990 for 29.90€
vads_currency	Code of the payment currency	978 for Euro

Table 10: Field list - Recurring payment by SEPA Direct Debit

2. Set **vads_page_action** to **REGISTER_PAY**.

3. Set **vads_action_mode** to **INTERACTIVE**.

   SEPA Direct Debit is not compatible with the SILENT mode.

4. Set the **vads_currency** field to **978** (Euro currency code).

   Only the Euro currency is accepted for this payment method.

5. Populate the **vads_amount** field with an amount in the smallest currency unit (in cents for Euro).

6. If you would like to pre-fill the form, add the optional fields of the table below to create your payment form:

Field name	Description	Value
vads_cust_title	Allows to specify the buyer's title.	Eg.: • **Mr** • **Mrs** • **Ms**
vads_cust_last_name	Allows to specify the buyer's last name.	Eg.: Dupont
vads_cust_first_name	Allows to specify the buyer's first name.	Eg.: Jean
vads_cust_cell_phone	Buyer cell phone number to receive the security code (by SMS). **Note**: The buyer cannot modify the pre-filled cell phone number on the payment page. Only GSM-users having recorded this phone number at the moment of subscription can receive an SMS with this access code for single use. If the phone number in unknown, the buyer will have to enter it on the payment page.	Eg.: 0612345678
vads_cust_country	Allows to specify the buyer's country code in compliance with the ISO 3166 standard.	Eg.: **FR**

Table 11: Field list - Recurring payment by SEPA Direct Debit

7. Set **vads_payment_cards** if you would like to pre-select the payment method on the merchant website.

   Note:

   - *If this list contains only one card type, the data entry page for this payment method will directly appear.*

     *Expected value for SEPA Direct Debit is **SDD**.*

   - *If there are more than one card types, the page with payment method selection will appear.*
   - *If this parameter is empty (recommended), the available payment methods (currencies, technical constraints, etc.) associated with the shop will be proposed.*

8. Add optional fields according to your requirements (see chapter **Using additional functions**).

9. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of a form:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_cust_email" value="name@example.com" />
<input type="hidden" name="vads_page_action" value="REGISTER_PAY" />
<input type="hidden" name="vads_site_id" value="70258842" />
<input type="hidden" name="vads_amount" value="2990" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_trans_date" value="20140919130128" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="96b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

7.5. Updating a mandate and making a payment

Prerequisites

Before updating a mandate, you must have registered it (vads_page_action = REGISTER) .

In order to update a recurring mandate and make a payment:

1. Use all the fields in the table below to create your payment form.

Field name	Description	Value
vads_site_id	Shop ID	Eg.: 12345678
vads_ctx_mode	Operating mode	**TEST** or **PRODUCTION**
vads_trans_date	Date and time of the payment form in UTC format	Eg.: 20140919130128
vads_action_mode	Acquisition mode for bank data.	**INTERACTIVE**
vads_page_action	Action required to **update** a mandate and perform a payment	**REGISTER_UPDATE_PAY**
vads_version	Version of the exchange protocol	**V2**
vads_cust_email	Buyer's e-mail address. **Note**: The buyer cannot modify the pre-filled e-mail address on the payment page. If the e-mail address in unknown, the buyer will have to enter it on the payment page.	Eg.: name@example.com
vads_identifier	Reminder of the payment token to debit	E.g.: 22e75ce992924f069279f42afe18da10
vads_amount	Payment amount (in the smallest currency unit)	Example: 2990 for 29.90€
vads_currency	Code of the payment currency	978 for Euro

Table 12: Field list - Recurring payment by SEPA Direct Debit

2. Set **vads_page_action** to **REGISTER_UPDATE_PAY** .

   Note:

   *Do not confuse **REGISTER_UPDATE_PAY** with **REGISTER_UPDATE**.*

   ***REGISTER_UPDATE** only allows to update a mandate without making a payment.*

3. Set **vads_action_mode** to **INTERACTIVE**.

   SEPA Direct Debit is not compatible with the SILENT mode.

4. Set the **vads_currency** field to **978** (Euro currency code).

   Only the Euro currency is accepted for this payment method.

5. Populate the **vads_amount** field with an amount in the smallest currency unit (in cents for Euro).

6. Populate the **vads_identifier** field with the token (identifier) that you obtained when registering the mandate.

7. If you would like to pre-fill the form, add the optional fields of the table below to create your payment form:

Field name	Description	Value
vads_cust_title	Allows to specify the buyer's title.	Eg.: • **Mr** • **Mrs** • **Ms**
vads_cust_last_name	Allows to specify the buyer's last name.	Eg.: Dupont

Field name	Description	Value
vads_cust_first_name	Allows to specify the buyer's first name.	Eg.: Jean
vads_cust_cell_phone	Buyer cell phone number to receive the security code (by SMS). **Note**: The buyer cannot modify the pre-filled cell phone number on the payment page. Only GSM-users having recorded this phone number at the moment of subscription can receive an SMS with this access code for single use. If the phone number in unknown, the buyer will have to enter it on the payment page.	Eg.: 0612345678
vads_cust_country	Allows to specify the buyer's country code in compliance with the ISO 3166 standard.	Eg.: **FR**

Table 13: Field list - Recurring payment by SEPA Direct Debit

8. Set **vads_payment_cards** if you would like to pre-select the payment method on the merchant website.

   Note:

   - *If this list contains only one card type, the data entry page for this payment method will directly appear.*

     Expected value for SEPA Direct Debit is ***SDD****.*
   - *If there are more than one card types, the page with payment method selection will appear.*
   - *If this parameter is empty (recommended), the available payment methods (currencies, technical constraints, etc.) associated with the shop will be proposed.*

9. Add optional fields according to your requirements (see chapter **Using additional functions**).

10. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of a form:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_cust_email" value="name@example.com" />
<input type="hidden" name="vads_identifier" value="22e75ce992924f069279f42afe18da10" />
<input type="hidden" name="vads_page_action" value="REGISTER_UPDATE_PAY" />
<input type="hidden" name="vads_site_id" value="70258842" />
<input type="hidden" name="vads_amount" value="2990" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_trans_date" value="20140919130128" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="89b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

8. USING ADDITIONAL FUNCTIONS

To obtain a customized form, you can use some optional features from the list below:

- Pre-filling the form
- Customizing the UMR (Unique Mandate Reference)
- Defining a different amount for the first installment(s)

Other functions are presented in the payment form implementation guide. They will help you easily create a payment form.

8.1. Customizing the UMR (Unique Mandate Reference)

By default the UMR is generated by the payment gateway.

However, the merchant can generate and transmit the value, taking into account the following requirements:

- The value must be transmitted in the field **vads_identifier**.
- The merchant must make sure that the UMR is unique.

  Any request for one-time or recurring direct debit made with an existing UMR will lead to an error and a technical error message will appear.
- Length: up to 35 characters.
- Authorized characters:
  - a b c d e f g h i j k l m n o p q r s t u v w x y z
  - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  - 0 1 2 3 4 5 6 7 8 9 / - ? : ( ) . , ' +
  - Space is not an authorized character.

8.2. Modifying the desired due date

By default, in the Back Office:

- The default capture delay is set to 0 days,
- The delay for pre-notification is set to 14 days.

With these settings:

- The pre-notification date is D,
- The desired due date is D+14,

These settings can be modified directly in the Back Office.

It can also be surcharged in the form by setting the parameter value of **vads_capture_delay** to a value superior to the delay of the pre-notification.

Example:

Value of the capture delay	Pre-notification date.	Desired due date
0	D	D+14
7	D	D+14
15	D+1	D+15

Table 14: Examples of values for the vads_capture_delay field

Warning, by modifying this value you directly influence the capture date and the pre-notification date.

8.3. Managing the return to the merchant website

At the end of payment, the buyer has the possibility to return to the merchant website via a **Return URL**.

Not to be confused with **Instant notification (IPN) URL** (see chapter **Managing the interaction with the merchant website**).

Defining the Return URLs

In the payment form, the merchant can override the configuration of the Back Office. To do so, the merchant can:

- Use 4 different URLs depending on the payment result:
  - Payment accepted.
  - Payment declined.
  - Payment abandoned.
  - Payment error.
- Or use one single URL independently of the payment result.

Defining the Return URLs depending on the payment result

1. Use the fields required for your use case (see chapter **Generating a payment form**) to create your payment form.

2. Use the optional fields presented in the table below to create a customized payment form.

   If no URL is specified in the form, the value populated in the Back Office will be used.

Field name	Description
vads_url_success	URL where the buyer will be redirected in case of an accepted payment after having clicked on "Return to shop".
vads_url_refused	URL where the buyer will be redirected in case of a declined payment after having clicked on "Return to shop".
vads_url_cancel	URL where the buyer will be redirected after having clicked on "Cancel and return to shop" before processing the payment
vads_url_error	URL where the buyer will be redirected in case of a processing error on the payment gateway.

3. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of a payment form with a definition of a Return URL depending on the payment result:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="3000" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20140526101407" />
<input type="hidden" name="vads_trans_id" value="239848" />
<input type="hidden" name="vads_url_cancel" value="http://demo.com/cancel.php" />
<input type="hidden" name="vads_url_error" value="http://demo.com/error.php" />
<input type="hidden" name="vads_url_refused" value="http://demo.com/refused.php" />
<input type="hidden" name="vads_url_success" value="http://demo.com/success.php" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="86b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

Defining a unique Return URL regardless of the payment result

1. Use the fields required for your use case (see chapter **Generating a payment form**) to create your payment form.

2. Use the optional field **vads_url_return** to define a redirection URL at the end of payment.

   If no URL is specified in the form, the value populated in the Back Office will be taken into account.

3. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of a payment form with a unique Return URL regardless of the payment result:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="3000" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20140526101407" />
<input type="hidden" name="vads_trans_id" value="239848" />
<input type="hidden" name="vads_url_return" value="http://demo.com/return.php" />
<input type="hidden" name="vads_version" value="V2" />
```

```
<input type="hidden" name="signature" value="86b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

Defining the method for receiving data

By default, the merchant website does not receive any parameters.

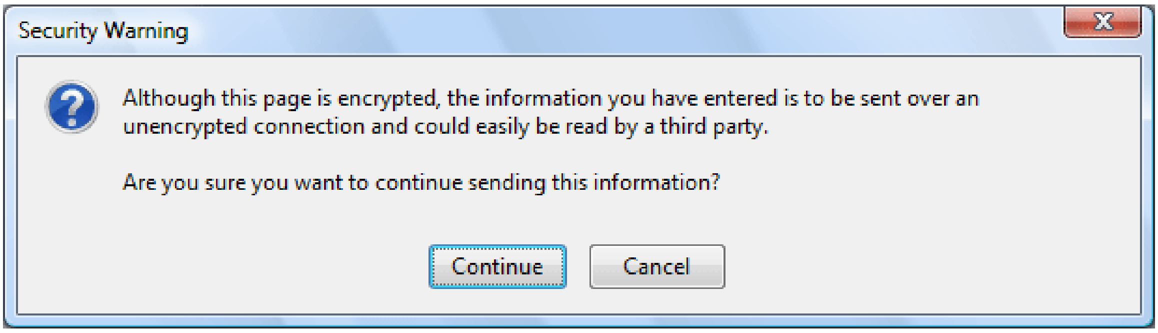
To retrieve information on the return page (tracking, statistics, customization of messages sent to the buyer, etc.), the merchant can override this configuration in the payment form.

1. Use the fields required for your use case (see chapter **Generating a payment form**) to create your payment form.

2. Use the optional field **vads_return_mode** to indicate the method for submitting data to the merchant website.

Value	Description
Absent, vide ou **NONE**	No information has been transmitted to the Return URL.
GET	The information has been transmitted in the URL of the return page.
POST	The information has been transmitted to the return page in an HTTP **POST** form.

The **GET** method allows to keep a notification message from appearing when the return is done from an **insecure environment (http)**.



Security Warning

Although this page is encrypted, the information you have entered is to be sent over an unencrypted connection and could easily be read by a third party.

Are you sure you want to continue sending this information?

3. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

<u>Example of a payment form with definition of the mode for data transmission</u>:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="3000" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_return_mode" value="GET" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20140526101407" />
<input type="hidden" name="vads_trans_id" value="239848" />
<input type="hidden" name="vads_url_return" value="http://demo.com/return.php" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="86b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

8.4. Defining a different amount for the first installment(s)

You wish to define a recurring payment for which the amount of the first installment would differ from the one set in the **vads_sub_amount** field.

Example: to define a recurring payment with the first 3 installments of 25 euros and the rest of the installments of 30 euros.

To do this:

1. Use the fields required for your use case subscription with a fixed date and amount) to create your payment form.

2. Use the fields below:

Field name	Value
vads_sub_currency	978
vads_sub_init_amount_number	3
vads_sub_init_amount	2500
vads_sub_amount	3000

Table 15: List of fields to define a different amount for the first installments

3. Compute the value of the signature field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of a payment form

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="2990" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_cust_country" value="FR" />
<input type="hidden" name="vads_cust_email" value="example@gmail.com" />
<input type="hidden" name="vads_cust_first_name" value="Paul" />
<input type="hidden" name="vads_cust_last_name" value="Juve" />
<input type="hidden" name="vads_cust_title" value="Mr" />
<input type="hidden" name="vads_page_action" value="REGISTER_SUBSCRIBE" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="91335531" />
<input type="hidden" name="vads_trans_date" value="20140716080441" />
<input type="hidden" name="vads_trans_id" value="362812" />
<input type="hidden" name="vads_validation_mode" value="0" />
<input type="hidden" name="vads_sub_currency" value="978" />
<input type="hidden" name="vads_sub_init_amount_number" value="3" />
<input type="hidden" name="vads_sub_init_amount" value="2500" />
<input type="hidden" name="vads_sub_amount" value="3000" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="86b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

9. COMPUTING THE SIGNATURE

To compute the signature you must have all the fields starting with **vads_** and the **certificate** (the certificate value allows to compute the numerical signature).

To compute the signature:

1. Sort the fields starting with **vads_** alphabetically.
2. Make sure that all the fields are encoded in UTF-8.
3. Concatenate the values by separating them with a "**+**".
4. Concatenate the result with the test or production certificate by separating them with a "**+**".
5. Apply the SHA-1 algorithm to obtain the signature value.

Example of parameters sent to the payment gateway:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="1524" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20090501193530" />
<input type="hidden" name="vads_trans_id" value="654321" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="606b369759fac4f0864144c803c73676cbe470ff"/>
<input type="submit" name="payer" value="Payer"/>
</form>
```

This example is analyzed as follows:

1. Sort the fields starting with **vads_** **alphabetically**:
   - vads_action_mode
   - vads_amount
   - vads_ctx_mode
   - vads_currency
   - vads_page_action
   - vads_payment_config
   - vads_site_id
   - vads_trans_date
   - vads_trans_id
   - vads_version
2. Concatenate these values by separating them with a"**+**":

   INTERACTIVE+1524+TEST+978+PAYMENT+SINGLE+12345678+20090501193530+654321+V2
3. Add the test certificate value at the end of the chain by separating it with a "**+**". In this example, the test certificate is **1122334455667788**:

   INTERACTIVE+1524+TEST+978+PAYMENT+SINGLE+12345678+20090501193530+654321+V2+1122334455667788
4. Apply the SHA-1 algorithm to the resulting chain.

   The result that must be transmitted to the signature field is: 606b369759fac4f0864144c803c73676cbe470ff; **606b369759fac4f0864144c803c73676cbe470ff**.

10. SENDING THE PAYMENT REQUEST

The buyer will be able to finalize his/her purchase once he/she is redirected to the payment page.

The buyer's browser must transmit the payment form data.

10.1. Redirecting the buyer to the payment page

The URL of the payment gateway is:

https://secure.payzen.eu/vads-payment/

Example of parameters sent to the payment gateway:

```
<form method="POST" action="https://secure.payzen.eu/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="2990" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_cust_country" value="FR" />
<input type="hidden" name="vads_cust_email" value="name@example.com" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20140526101407" />
<input type="hidden" name="vads_trans_id" value="362812" />
<input type="hidden" name="vads_validation_mode" value="0" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="86b2a17b9a5fcefb6c0120c57b25ec86ad1704ee"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

10.2. Processing errors

If the payment gateway detects an error while receiving the form, an error message will appear and the buyer will not be able to proceed to the payment.

In TEST mode

The message indicates to the buyer that a technical problem occurred and proposes to follow a link to the FAQ to help fix the error.

In PRODUCTION mode

The message simply indicates to the buyer that a technical problem occurred.

In both cases the merchant receives a notification e-mail.

It contains:

- the source of the error,
- a link to FAQ to facilitate its analysis,
- all the fields of the form.

The FAQ is available here: *https://secure.payzen.eu/html/error_code/*

11. ANALYZING THE PAYMENT RESULT

To process the payment result the merchant website must have a separate page with a script (e.g. analyze_payment.php).

This page will be called automatically after each payment (whether it has been accepted or declined): the parameters linked to the payment result are sent in POST mode by the payment gateway.

Prerequisites:

- URL of the page that analyzes the payment result must be specified in the Back Office (see chapter).
- The merchant has to make sure that this URL is available from the payment gateway without redirection.

  Using redirection leads to losing data presented in POST.
- In case some restrictions are set up by the merchant, the **194.50.38.0/24** IP address range must be authorized.
- HTML should not be visible on the page.

  Access to images or CSS may slow down the exchange between the payment gateway and the merchant website.

  Furthermore, the payment gateway systematically reads the 512 first characters returned by the merchant website.

  These characters can be viewed in the transaction history.
- Avoid integrating time-consuming tasks, such as invoice generation or sending e-mails in the script.

  The processing time has a direct influence on how long it takes to display the payment summary page. The longer the processing of the notification, the greater the delay for displaying the page.

  After 35s the payment gateway will consider that the call has failed (timeout).

Failed notification (IPN)

In case the call to IPN fails, a notification e-mail is sent to the address specified in the Back Office (see chapter **Setting up notifications**).

The e-mail contains:

- the HTTP code of the encountered error,
- analysis parts depending on the error,
- instructions for resending the notification from the Back Office.

Writing the processing script

The processing script must include at least the following steps:

- retrieve the field list sent with the response in POST mode,
- compute the signature,
- compare the computed signature with the received signature,
- analyze the nature of the notification,
- retrieve the payment result.

The script may check the order status (or any information of your choice) to see if it has not been already updated.

Once these steps are completed, the script can update the database (new order status, stock update, registration of payment information, etc.).

11.1. Retrieving data returned in the response

The data returned in the response depends on the parameters sent in the payment form, on the payment type and on the settings of your shop. This data constitutes a field list. Each field contains a response value. The field list can be updated.

The data is always sent by the payment gateway in **POST** mode.

The first step consists in retrieving the content received in POST mode.

Examples:

- in PHP, data is stored in the super global variable **$_POST**,
- in ASP.NET (C#), you must use the **Form** property of the **HttpRequest** class,
- in Java, you must use the **getParameter** method of the **HttpServletRequest** interface,

  The script will have to create a loop to retrieve all the transmitted fields.

Retrieving data at the moment of instant notification

The data is always sent by the payment gateway via **POST**.

The transmitted information includes:

- the **vads_url_check_src** field defines the applied notification rule:

Value	Applied rule
PAY	End of payment notification URL.
BO	Execution of the notification URL from the Back Office.
BATCH_AUTO	The notification URL called at the moment of the authorization request for payments with the "Waiting for authorization" status.
REC	The notification URL called for installment payments (subscription option).
MERCH_BO	IPN called during an operation performed from the Back Office.

Table 16: Values of the vads_url_check_scr field

You must test its value to identify the nature of the notification.

- the **vads_hash** field representing a unique key associated with the transaction.

These two fields will not be sent at the moment of return to the shop.

Retrieving data at the moment of return to shop

By default, when the buyer returns to the merchant website, no parameters will be transmitted by the buyer's browser.

However, according to the configuration used, (see chapter **Defining data transmission mode**), it will be possible to retrieve the data:

- either in GET mode: the data is presented in the URL as follows: ?field1=value1&field2=value2.
- or in POST: the data is sent in a POST form.

The data transmitted to the browser is the same as during notifications (IPN), except the **vads_url_check_src** and **vads_hash** fields.

11.2. Computing the signature

The signature is computed by following the same procedure as for creating the payment form.

All the received fields must be included.

To compute the signature:

1. Take all the fields starting with **vads_**.
2. Sort these fields alphabetically.
3. Make sure that all the fields are encoded in UTF-8.
4. Concatenate the result with the test or production certificate by separating them with a "**+**".
5. Apply the SHA-1 algorithm to obtain the signature value.

11.3. Comparing signatures

To ensure the integrity of the response, you must compare the value of the **signature** field received in the response with the one computed previously.

If the signatures match,

- you may consider the response as safe and proceed with the analysis,
- if they do not, the script will have to throw an exception and warn the merchant (see chapter **Processing errors** of the Pyament Form Integration Guide available on our web site *https://payzen.io/*).

The signatures may not match because of:

- an implementation error (error in your calculation, problem with UTF-8 encoding, etc.),
- an error in the value of the certificate or in the **vads_ctx_mode** field value (frequent issue when going to live mode),
- a data corruption attempt.

11.4. Analyzing the nature of notification

The **vads_url_check_src** field allows to differentiate the notifications according to their triggering event (creation of a transaction, new notification sent by the merchant from the Back Office, etc.):

- mandate registration (whether the subscription has been defined or not),
- an installment payment,
- a new notification sent by the merchant from the Back Office.

It specifies the applied notification rule:

Value	Applied rule
PAY	The PAY value will be sent in the following cases: • request for mandate registration (REGISTER), • request for mandate registration with a definition of a subscription (REGISTER_SUBSCRIBE), • immediate payment (or first installment payment), • Only if the merchant has configured the rule for **Instant Payment Notification URL at the end of payment**. • the payment that has been abandoned or canceled by the buyer, only if the merchant has configured the rule for **Instant Payment Notification URL on cancellation**.
BO	Execution of the notification from the Back Office (right-click on a transaction > **Execute the IPN URL**).
BATCH_AUTO	The BATCH_AUTO value will be sent in the following cases: • deferred payment with a capture due date before/after the authorization expiration date (see the Authorization validity period section), • refund payments (except the first one), Only if the merchant has configured the rule for **Instant Payment Notification URL on batch authorization**. The notification will be sent with the authorization request for payments with "Awaiting authorization" status.
REC	The REC value is sent only for subscription payments if the merchant has configured the rule for **Instant Payment Notification URL when creating recurring payments**.
MERCH_BO	The MERCH_BO value will be sent in the following cases: • during operation performed from the Back Office (refund, modification, validation, duplication), only if the merchant has configured the following notification rule: **Instant Payment Notification URL on an operation coming from the Back Office** • during cancellations made from the Back Office only if the merchant has configured the following notification rule: **Instant Payment Notification URL on cancellation**

Table 17: Values of the vads_validation_mode field

After checking its value, the script will be able to process differently depending on the nature of the notification;

For example:

If **vads_url_check_src** is set to **PAY** or **BATCH_AUTO**, the script will update the order status, etc.

If **vads_url_check_src** is set to **REC**, the script will retrieve the subscription reference and will increment the number of the expired installment payments in case the payment has been accepted, etc.

11.5. Processing the response to a request for a subscription with a fixed date and amount

1. Analyze the result of mandate registration by testing the value of the **vads_identifier_status** field.

Value	Description
CREATED	Mandate is successfully registered.
NOT_CREATED	Mandate has not been created and cannot be viewed in the Back Office. No direct debits will be made.
UPDATED	Mandate has been successfully updated.
ABANDONED	Mandate registration has been abandoned by the debtor (buyer). Mandate has not been created and cannot be viewed in the Back Office. No direct debits will be made.

Table 18: Values of the vads_url_check_scr field

2. Retrieve the mandate details:

Field	Description
vads_identifier	Returns the Unique Mandate Reference Number. This value will be used later for making a payment, creating a subscription, exchanging files, etc.
vads_card_number	Returns the IBAN and BIC, separated by a "_".
vads_expiry_month	Returns the expiration month of the mandate
vads_expiry_year	Returns the expiration year of the mandate

Table 19: Mandate details

3. Analyze the result of mandate creation by testing the value of the **vads_recurrence_status** field.

Value	Description
CREATED	A subscription has been successfully created. Subscription details are visible in the Back Office.
NOT_CREATED	The subscription has not been created and is not visible in the Back Office. No direct debits can be made.
ABANDONED	Mandate creation has been abandoned by the debtor (buyer). The subscription has not been created and is not visible in the Back Office. No direct debits can be made.

Table 20: Values of the vads_url_check_scr field

4. Retrieve the subscription details:

Field name	Description
vads_subscription	Subscription ID
vads_sub_effect_date	Subscription start date.
vads_sub_amount	Amount of each installment (cents) except the ones that will be eventually defined by the **vads_sub_init_amount_number**
vads_sub_currency	Numeric code of the euro currency (978).
vads_sub_init_amount	Amount of the first installments.
vads_sub_init_amount_number	Number of installments for which the amount **vads_sub_init_amount** should be applied.
vads_sub_desc	Rule for recurring payments to apply according to the iCalendar RFC5545 specification.

Table 21: Parameter list - Subscription details

Exemple of a form:

```
vads_amount =
vads_auth_mode = MARK
vads_auth_number =
vads_auth_result =
vads_capture_delay = 0
vads_card_brand = SDD
vads_card_number = FR7617515900001234567890135_CEPAFRPP751
vads_payment_certificate =
vads_ctx_mode = TEST
vads_currency = 978
vads_effective_amount =
vads_site_id = 91335531
vads_trans_date = 20140925090433
vads_trans_id =
vads_validation_mode = 0
vads_version = V2
vads_warranty_result = NO
vads_payment_src = EC
vads_cust_email = lastname@example.com
vads_cust_last_name = Lastname
vads_contract_used = FR7630002005701234567890158
vads_recurrence_status = CREATED
vads_identifier_status = CREATED
vads_expiry_month = 9
vads_expiry_year = 2017
vads_pays_ip = FR
vads_identifier = 20140925fRbwlu
vads_subscription = 20140925FRwRw9
vads_threeds_enrolled =
vads_threeds_cavv =
vads_threeds_eci =
vads_threeds_xid =
vads_threeds_cavvAlgorithm =
vads_threeds_status =
vads_threeds_sign_valid =
vads_threeds_error_code = 4
vads_threeds_exit_status = 4
vads_result = 00
vads_extra_result =
vads_card_country =
vads_language = fr
vads_hash = 8c094f21dca9f0279108a87efc0d5bc49337a8613451271e201d1cfcdbbbbeb3
vads_url_check_src = PAY
vads_action_mode = INTERACTIVE
vads_payment_config = SINGLE
vads_page_action = REGISTER_SUBSCRIBE
vads_sub_amount = 2990
vads_sub_currency = 978
vads_sub_desc = RRULE:FREQ=MONTHLY;BYMONTHDAY= 1
vads_sub_effect_date = 20141101
signature = a5026f230479eb17739609cb98c0e70e6516a006
```

11.6. Analyzing the result of an installment

1. Identify the mandate and the subscription that correspond to the installment by testing the following fields:

Field	Description
vads_identifier	Mandate reference (UMR)
vads_subscription	Subscription ID
vads_sequence_number	Recurrence number of the subscription

Table 22:

2. Analyze the payment result by testing the value of the **vads_trans_status** field

Value	Description
ABANDONED	The payment has been abandoned by the buyer. The transaction has not been created and **cannot be viewed in the Back Office**.
AUTHORISED	A pre-notification has been sent. The direct debit is waiting to be captured.
AUTHORISED_TO_VALIDATE	The direct debit request has been made in manual validation mode. The merchant must manually validate the transaction in the Back Office to allow the mandate to be sent to the bank. The pre-notification will only be sent after the transaction has been validated. An automatic e-mail is processed every day, several times per day (every 3 hours) that does not require any intervention on your behalf. The transaction can be validated as long as the capture date has not passed. If the capture date has passed, the payment status changes to **Expired**. The expired status is final.
WAITING_AUTHORISATION	The pre-notification has not been sent yet. The direct debit request has been made with a capture delay greater (strictly) to 14 days while the pre-notification delay in the contract is 14 days.

Table 23: Values of the vads_url_check_scr field

3. Retrieve the transaction details:

Field name	Description
vads_trans_id	Returns the transaction identifier.
vads_presentation_date	Returns the desired due date.
vads_amount	Payment amount (in cents).
vads_currency	Numeric code of the euro currency (978).

Table 24: Parameter list - Transaction details

11.7. Processing the response to the request for mandate registration

1. Analyze the result of mandate registration by testing the value of the **vads_identifier_status** field.

Value	Description
CREATED	Mandate is successfully registered.
NOT_CREATED	Mandate has not been created and cannot be viewed in the Back Office. No direct debits will be made.
UPDATED	Mandate has been successfully updated.
ABANDONED	Mandate registration has been abandoned by the debtor (buyer). The mandate has not been created and cannot be viewed in the Back Office. No direct debits will be made.

Table 25: Values of the vads_url_check_scr field

2. Retrieve the mandate details:

Field	Description
vads_identifier	Returns the Unique Mandate Reference Number. This value will be used for exchanging files.
vads_card_number	Returns the IBAN and BIC, separated by a "_".
vads_expiry_month	Returns the expiration month of the mandate
vads_expiry_year	Returns the expiration year of the mandate

Table 26: Mandate details

Exemple of a form:

```
vads_auth_mode = MARK
vads_auth_number =
vads_auth_result =
vads_capture_delay = 0
vads_card_brand = SDD
vads_card_number = FR7617515900001234567890135_CEPAFRPP751
vads_payment_certificate =
vads_ctx_mode = TEST
vads_currency = 978
vads_effective_amount =
vads_site_id = 91335531
vads_trans_date = 20140925082339
vads_trans_id =
vads_validation_mode = 0
vads_version = V2
vads_warranty_result = NO
vads_payment_src = EC
vads_cust_email = firstname.lastname@exemplecom
vads_cust_title = M.
vads_cust_last_name = Lastname
vads_contract_used = FR7630002005701234567890158
vads_identifier_status = CREATED
vads_expiry_month = 9
vads_expiry_year = 2017
vads_pays_ip = FR
vads_identifier = 20140925fGwQ01
vads_threeds_enrolled =
vads_threeds_cavv =
vads_threeds_eci =
vads_threeds_xid =
vads_threeds_cavvAlgorithm =
vads_threeds_status =
vads_threeds_sign_valid =
vads_threeds_error_code = 4
vads_threeds_exit_status = 4
vads_result = 00
vads_extra_result =
vads_card_country =
vads_language = fr
vads_hash = 9da7bc10ee4ca24aa5453799ceb62b785c7be7b69e0d0f66cc9fdd467d9cb703
vads_url_check_src = PAY
vads_action_mode = INTERACTIVE
vads_payment_config = SINGLE
vads_order_id = cyv28329
vads_page_action = REGISTER
```

```
signature = b6996bdc20cf42e60c7c381c7841f515876a91de
```

11.8. Processing the response to a mandate update and analyzing the payment result

1. Analyze the result of the mandate update by testing the value of the **vads_identifier_status** field.

Value	Description
CREATED	Mandate is successfully registered.
NOT_CREATED	Mandate has not been created and cannot be viewed in the Back Office. No direct debits will be made.
UPDATED	Mandate has been successfully updated.
ABANDONED	Mandate registration has been abandoned by the debtor (buyer). Mandate has not been created and cannot be viewed in the Back Office. No direct debits will be made.

Table 27: Values of the vads_identifier_status field

2. Retrieve mandate details:

Field	Description
vads_identifier	Returns the token (Unique Mandate Reference). This value will be used later to make a payment, create a subscription, exchange files, etc.
vads_card_number	Returns the IBAN and BIC, separated by a "_".
vads_expiry_month	Returns the expiration month of the mandate
vads_expiry_year	Returns the expiration year of the mandate

Table 28: Mandate details

3. Analyze the transaction result by testing the value of the **vads_trans_status** field:

Value	Description
ABANDONED	The payment has been abandoned by the buyer. The transaction has not been created and **cannot be viewed in the Back Office**.
AUTHORISED	A pre-notification has been sent. The direct debit is waiting to be captured.
AUTHORISED_TO_VALIDATE	The direct debit request has been made in manual validation mode. The merchant must manually validate the transaction in the Back Office to allow the mandate to be sent to the bank. The pre-notification will only be sent after the transaction has been validated. An automatic e-mail is processed every day, several times per day (every 3 hours) that does not require any intervention on your behalf. The transaction can be validated as long as the capture date has not passed. If the capture date has passed, the payment status changes to **Expired**. The expired status is final.
WAITING_AUTHORISATION	The pre-notification has not been sent yet. The direct debit request has been made with a capture delay greater (strictly) to 14 days while the pre-notification delay in the contract is 14 days.

Table 29: Values of the vads_url_check_scr field

4. Retrieve the transaction details:

Field name	Description
vads_trans_id	Returns the transaction identifier.
vads_presentation_date	Returns the desired due date.
vads_amount	Payment amount (in cents).
vads_currency	Numeric code of the euro currency (978).

Table 30: Parameter list - Transaction details

11.9. Analyzing the result of a one click payment

1. Identify the mandate that correspond to the payment by testing the following field:

Field	Description
vads_identifier	Mandate reference (UMR)

2. Analyze the payment result by testing the value of the **vads_trans_status** field

Value	Description
ABANDONED	The payment has been abandoned by the buyer. The transaction has not been created and **cannot be viewed in the Back Office**.
AUTHORISED	A pre-notification has been sent. The direct debit is waiting to be captured.
AUTHORISED_TO_VALIDATE	The direct debit request has been made in manual validation mode. The merchant must manually validate the transaction in the Back Office to allow the mandate to be sent to the bank. The pre-notification will only be sent after the transaction has been validated. An automatic e-mail is processed every day, several times per day (every 3 hours) that does not require any intervention on your behalf. The transaction can be validated as long as the capture date has not passed. If the capture date has passed, the payment status changes to **Expired**. The expired status is final.
WAITING_AUTHORISATION	The pre-notification has not been sent yet. The direct debit request has been made with a capture delay greater (strictly) to 14 days while the pre-notification delay in the contract is 14 days.

Table 31: Values of the vads_url_check_scr field

3. Retrieve the transaction details:

Field name	Description
vads_trans_id	Returns the transaction identifier.
vads_presentation_date	Returns the desired due date.
vads_amount	Payment amount (in cents).
vads_currency	Numeric code of the euro currency (978).

Table 32: Parameter list - Transaction details

12. MANAGING YOUR DIRECT DEBITS VIA A FILE SHARING SERVICE

To create a subscription with flexible amounts and due dates, you must:

- Register a recurring mandate.

The steps for generating, uploading and analyzing a payment file are described in the next chapters.

12.1. Generating a payment file

The payment request file will consists of a series of records.

A "record" is a list of parameters separated by ";" that end with a carriage return. Thus, there is one record on every line.

This format type is generally called "CSV" (Comma Separated Values), and is notably recognized by MS Excel.

The file will have the following structure:

- A header record containing transmission details.
- A variable amount of records containing details of payments to be made.
- A trailer record that allows to verify the consistency of the transmitted file.

Example:

A test file was submitted on 30/11/2016 for the shop 12345678. It contains two payment requests with the following parameters:

First installment:

- UMR: ‘MY-UMR-TEST-1’
- Transaction number: 000001
- Amount: €32.99

Second installment:

- UMR: ‘MY-UMRTEST-2’
- Transaction number: 000002
- Amount: €7.90

1. The name of the file must respect the format <YYYYMMDD>.<siteID>.PAY.REQ.<z>.<xx>.

   Where:

   - **<YYYYMMDD>** represents the date when the file was generated.
   - **<siteID>** represents the shop number that has been assigned to you (vads_site_id).
   - **<z>** takes either the value "T" for a file in TEST mode, or "P" for a file in PRODUCTION mode.
   - **<xx>** represents the sequential number of the file within a day, in 2 numeric characters.

   The name of the file in the example is 20161130.12345678.PAY.REQ.T.01

2. Populate the header record with the following values:

Pos.	Description	Format	Values	Equivalent
01	Record code	n2	**00**	

Pos.	Description	Format	Values	Equivalent
02	File type		**PAY**	
03	File version	n2	**02**	
04	Shop ID	n8	**12345678**	vads_site_id
05	Environement (TEST or PRODUCTION)		**TEST**	vads_ctx_mode
06	Creation date	n8	In the YYYYMMDD format	
07	Time of creation	n6	In the HHMMSS format	
08	Requested execution date.	n8	Reserved for future use. Do not populate	

3. Populate each detail record respecting the format below:

Pos.	Description	Format	Values	Equivalent
01	Record code	n2	**02**	
02	Sequence number of the detail record in the file.	n..6	Starts at 1	
03	Transaction date	n8	In the YYYYMMDD format	vads_trans_date
04	Transaction time	n6	In the HHMMSS format	vads_trans_date
05	05	n..6	Unique on a given day and shared with other payment modes	vads_trans_id
06	Transaction type	a2	**CD**	
07	Amount	n..12	In cents.	vads_amount
08	Euro currency code	n3	**978**	vads_currency
09	Expected due date	n8	In the YYYYMMDD format, leave empty to indicate the current date	
10	Validation mode	n..1	0 for automatic validation 1 for manual validation Empty for using the default validation mode	vads_validation_mode
11	Unique Mandate Reference	ans..50	UMR to use for the direct debit	vads_identifier
12	The commercial contract to use for the payment.	ans..250	Leave empty for using the default contract.	vads_contract_used
13	Order reference	ans..32	Leave empty if not used	vads_order_id
14	Details of order 1	ans..255	Leave empty if not used	vads_order_info
15	Details of order 2	ans..255	Leave empty if not used	vads_order_info2
19	Details of order 3	ans..255	Leave empty if not used	vads_order_info3

4. Populate the trailer record respecting the format below:

Pos.	Description	Format	Values	Equivalent
01	Record code	n2	**01**	
02	Number of transmitted detail records	n..6	**2**	

File example:

File name: 20161130.12345678.PAY.REQ.T.01

```
00;PAY;02;12345678;TEST;20161130;102008;
02;1;20161130;102008;000001;CD;3299;978;;;MY-UMRTEST-1;;;;;
02;2;20161130;102008;000002;CD;790;978;;;MY-RUM-TEST-2;;;;;
01;2
```

To assist you in integration, a validation interface is available in the Back Office via the menu: **Management** > **Test recurring payments** > **Payment request**.

It allows:

- To verify the structure of the file before submitting it
- To import the TEST payment file
- To create TEST payments on the basis of the imported file

12.2. Uploading payment files

The payment gateway processes the payment files transferred by the merchant website twice **every day**: once at **7 a.m.** and once at **1 p.m.**

It is recommended to upload the files to the FTP server before **6.50 a.m.** for the first analysis and **12.50 p.m.** for the second one.

1. Connect to the FTP server:

   In SFTP:

   - Host: *sftp://vadftp.lyra-network.com*
   - port: 222

   In FTPS:

   - Host: *ftps://vadftp.lyra-network.com*
   - port: 990

   It is advised to open the range 20220 to 20229 for "data" connexions.

   Login details were sent to you by e-mail at the moment of subscription.

2. Upload your file into the folder **Shop ID > upload**.

12.3. Downloading the return file

The payment gateway processes the payment files transferred by the merchant website twice **every day**: once at **7 a.m.** and once at **1 p.m.**

1. Connect to the FTP server.

2. Select the folder **Shop ID > id**.

   The response file will be available between **8 a.m.** and **9 a.m.** and between **2 p.m.** and **3 p.m.**

   The name of the associated response file is as follows:

   <YYYYMMDD>.<siteID>.PAY.ANS.<z>.<xx>

   Where:

   - **<YYYYMMDD>** represents the date when the file was generated.
   - **<siteID>** represents the shop number that has been assigned to you.
   - **<z>** takes either the value "T" for a file in TEST mode, or "P" for a file in PRODUCTION mode.
   - **<xx>** represents the sequential number of the file within a day, in 2 numeric characters.

12.4. Analyzing the return files

There are two categories of errors:

- formatting errors and/or field value errors,
- errors related to the execution of payments.

The platform processes files in two stages:

- the first stage serves to verify the integrity of the file and to detect errors of format and/or values. In case of error, the file is rejected completely.

  During this first stage, only global file checks are performed. The more detailed errors are processed during the second stage.

- During the second stage, payments are executed line by line. If an error is detected in one of the parameters (e.g.: id not found, incorrect currency, etc.), the processing code will get a value of 30 in the detailed record and an additional field will indicate the position of the data in question. In case of refusal of a payment, the source of the error will be indicated in the detailed record.

In the first error case, the file with the result will contain only a header record with the error and the trailer record. The merchant will have to correct and submit the file again after modifying its name.

In the second case, the result file will contain the same amount of detail records as the request file. Only the payments with an error detected in one of the parameters will have to be submitted one more time by the merchant.

If there is an issue on the platform which does not allow it to process the error in time, they will be analyzed when the platform is functioning properly again. Therefore, there is no need to resubmit the unprocessed payments in a new file.

Here is a response to our sample file 20161130.12345678.PAY.REQ.T.01

The name of the response file is 20161130.12345678.PAY.ANS.T.01.

It contains:

```
00;PAY;02;0;;12345678;TEST;20161130;102008;20140922;141349
02;1;20161130;102008;000001;CD;3299;978;3299;978;20140922;0;MY-UMR-TEST-1;
FR7630002005701234567890158;;;;;00;0;;FULL;20140922;141350;00;
FR7630002005701234567890158_CRLYFRPP;20170922
02;2;20161130;102008;000002;CD;790;978;790;978;20140922;0;MY-UMR-TEST-2;
FR7630002005701234567890158;;;;;00;0;;FULL;20140922;141351;00;
FR7630002005701234567890158_CRLYFRPP;20170922
01;2;2;0
```

Format of the header:

Pos.	Description	Format	Values	Data dictionary equivalence
01	Record code	n2	**00**	
02	File type		**PAY**	
03	File version	n2	**02**	
04	Return code of the first stage of processing	n1	0: Processing completed 1: Format error 2: Value error	
05	Information about the error	ans..255	Additional information about the error, empty if successfully completed.	

Pos.	Description	Format	Values	Data dictionary equivalence
06	Shop ID	n8	Same as the value used in the request, if valid.	vads_site_id
07	Environment		Same as the value used in the request, if valid.	vads_ctx_mode
08	Creation date	n8	Same as the value used in the request, if valid.	
09	Time of creation	n6	Same as the value used in the request, if valid.	
10	Date of end of processing In the YYYYMMDD format	n8	**20161130**	
11	Hour of end of processing	n6	**130019**	

Format of detailed record:

Pos.	Description	Format	Values	Data dictionary equivalence
01	Record code	n2	**02**	
02	Sequence number	n..6	Same as the value used in the request, if valid	
03	Transaction date (same as the value used in the request, if valid)	n8	Same as the value used in the request, if valid	vads_trans_date
04	Transaction time (same as the value used in the request, if valid)	n6	Same as the value used in the request, if valid	vads_trans_date
05	Transaction ID (same as the value used in the request, if valid)	n..6	Same as the value used in the request, if valid	vads_trans_id
06	Transaction type (same as the value used in the request, if valid)	a2	Same as the value used in the request, if valid	vads_site_id
07	Amount (same as the value used in the request, if valid)	n..12	Same as the value used in the request, if valid	vads_amount
08	Currency	n3	Same as the value used in the request, if valid	
09	Amount converted into Euro	n..12	Same as the value used in the request, if valid	
10	Counter currency	n3	Same as the value used in the request, if valid	
11	Due date	n8	Same as the value used in the request, if valid	
12	Validation mode	n1	0 if automatic validation 1 if manual validation	vads_validation_mode
13	UMR	ans..50	Same as the value used in the request, if valid	vads_identifier
14	Payment method number (Card number or IBAN...)	n7	Payment method number used to process this request	
15	Order reference	ans..32	Same as the value used in the request, if valid	vads_order_id
16	Information about transaction 1	ans..255	Same as the value used in the request, if valid	vads_order_info
17	Information about transaction 2	ans..255	Same as the value used in the request, if valid	vads_order_info2
18	Information about transaction 3	ans..255	Same as the value used in the request, if valid	vads_order_info3
19	Return code of processing		All the codes are available in the data dictionary. Examples: 00: Success	vads_result

Pos.	Description	Format	Values	Data dictionary equivalence
			30: Parameter error. Additional information in provided in field 25 96: Technical error	
20	Authorization return code	n..2	Code returned by the authorization server. All the codes are available in the data dictionary. Empty if not applicable.	vads_auth_result
21	Authorization number returned by the bank server.	n6	non applicable	vads_auth_number
22	Authorization mode		**FULL**	vads_auth_mode
23	Authorization date	n8	In the YYYYMMDD format	
24	Authorization time	n6	In the HHMMSS format	
25	Additional return code	ans..255	Contains the return of risk management (if subscribed), or the position of the incorrect field in case of error 30. All the codes are available in the data dictionary. Empty if not applicable.	vads_extra_result
26	Payment method number (Card number or IBAN and BIC...)	an..36	Extract of the payment method number (eg.: IBAN_BIC)	vads_card_number
27	Mandateexpiration date	n8	In the AAAAMMJJ format Empty if not applicable.	

Format of the trailer record:

Pos.	Description	Format	Values
01	Record code	n2	**01**
02	Total number of transmitted detailed records	n..6	
03	Number of successful payments	n..6	
04	Number of failed payments	n..6	

13. TRANSACTION PROCESSING

There are several operations related to SEPA Direct Debit (SDD) transactions you can perform from your Back Office.

In the list of transactions:

1. Select a SEPA direct debit.
2. Right-click to view the list of authorized operations.

The following chapters provide details on:

- Unavailable operations.
- Authorized operations.

13.1. Identifying unavailable operations

The following operations, in case of SDD payments, are not allowed:

- **Duplication**
- **Modification**

13.2. Identifying authorized operations

The authorized operations are:

- **Cancel**
- **Validate**
- **Edit the order reference**

  Allows to modify the order number.
- **Send the Instant Payment Notification URL**

  Allows to return a notification at the end of payment to the merchant website when the initial call has failed.
- **Resend the transaction confirmation e-mail**

  Allows to send payment confirmation e-mails to the merchant or the buyer.
- **Refund by wire transfer**

  Allows to proceed to refund by wire transfer from the Back Office.

  To do so, the SCT Service needs to be activated on your EBICS connection. Please contact your acquirer.
- **Cancel**

Manual cancellation of a recurring payment via the Back Office

1. Connect to your Back Office: *https://secure.payzen.eu/vads-merchant/*
2. Select **Management > Recurring payments**.
3. Search for the recurring payment to cancel.
4. Select this recurring payment with a right-click.
5. Select **Terminate**.

Automatic cancellation of a recurring payment via Web Services

- **Web Services V5**

Use the operation **cancelSubscription** to cancel a recurring payment on a specific date.

For more information, refer to the Web Services Technical Implementation Guide available in our online documentation archive:

https://payzen.io/

14. ACTIVATING THE SHOP IN PRODUCTION MODE

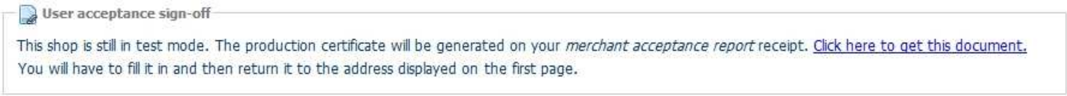
This chapter explains how you can:

- Generate the production certificate,
- Switch your merchant website to live mode,
- Make a first payment in production mode,
- Generate a new production certificate (in case a problem occurs).

14.1. Going to live mode

To go to live (production) mode, ask the customer service of your de payment gateway to generate your production certificate. To do this, you need to provide a document entitled **Merchant acceptance report**.

1. Go to your Back-Office.
2. Select **Settings > Shop > Certificates** tab.



User acceptance sign-off

This shop is still in test mode. The production certificate will be generated on your *merchant acceptance report* receipt. Click here to get this document. You will have to fill it in and then return it to the address displayed on the first page.

3. Click on the link **this address**.
4. Download the document entitled **Merchant acceptance report**.
5. Fill in all the required information.
6. Date and sign this **Merchant acceptance report** and send it to the address or the fax number indicated in the document.
7. An e-mail will be sent to the administrator contact to confirm the generation of the production certificate by the administrative service.
8. The production certificate will be available and visible in the **Certificates** tab.
   The value of the certificate will be masked for confidentiality purposes after a first successful payment.

14.2. Shifting your merchant website to production mode

Going to production mode:

- The variable **vads_ctx_mode** must be set to **PRODUCTION** in your form.
- The variable that contains your certificate must be modified with the production certificate.
- Enter the correct IPN in PRODUCTION mode by going to **Settings > Shop > Configuration** tab.

14.3. Making a first payment in production mode

It is recommended for the merchant to make a real transaction to verify that all of the components of the production environment function properly. Afterwards, you will be able to cancel this transaction in the Back Office of the payment gateway.

The Back Office is accessible at the following address: *https://secure.payzen.eu/vads-merchant/*.

From now on, transactions are available via the **Management** > **Transactions** menu, in the left top corner of the interface.

Verify that the IPN provided in the Back Office functions correctly without clicking on the **Return to shop** button.

14.4. Regenerating the production certificate

In case the production certificate is lost or corrupted, the merchant can generate a new one in the Back Office. To do so:

1. In the Back Office, select **Settings > Shop > Certificates** tab.
2. Click on **Regenerate**.

15. DATA DICTIONARY

The fields and their formats are listed in tables:

- **Name**: indicates the name of the parameter as it appears in the HTTP request.
- **Format**: indicates the data format as follows:

Data type	Description
a	Alphabetical characters (from 'A' to 'Z' and from 'a' to 'z')
n	Digital characters
s	Special characters
an	Alphanumeric characters
ans	Alphanumeric and special characters
3	The fixed length is 3 characters
..12	Flexible length - up to 12 characters

Table 33: List of fields and formats

- **Example**: represents an example of correct data encoding.
- **Error code**: in case there is an error in displaying the fields between the merchant website and the payment gateway, the payment gateway indicates the incorrect parameter in the **vads_extra_result** field using a numerical code.

ERROR CODE:

The error code is filled in when an incorrect payment form is being submitted.

- In TEST mode, the code appears on the payment page.
- In PRODUCTION mode, a notification message is sent stating the error code and the name of the incorrect parameter.

<u>Example</u>: Error 09 corresponds to an amount error. The sent amount does not respect the required format.

Viewing parameters sorted by category

Go to the desired category to obtain the list of related parameters.

- Information about 3DS authentication.
- Subscription details.
- Buyer details.
- Payment method details.
- Order details.
- Shipping details.
- Technical details.
- Transaction details.
- Information about the donation transactions.
- Payment page customization.
- Automatic redirection.

Technical details

Field name	Format	Description
signature	an40	Permits to check the integrity of the request.
vads_action_mode	string (enum)	Defines the acquisition mode of the credit card details.
vads_contrib	ans..128	Indicates the name of the plugin used during the payment (Joomla, osCommerce, etc.).
vads_ctx_mode	string (enum)	Defines the mode of interaction with the payment gateway.
vads_extra_result	n2	Optional code of the response. Its meaning depends on the value entered in **vads_result**.
vads_hash	an64	A unique key sent only within the IPN.
vads_page_action	string (enum)	Defines the action to be performed.
vads_payment_error	n..3	Error code for declined payments.
vads_result	n2	General return code of the payment outcome.
vads_site_id	n8	Shop ID.
vads_url_check	ans..127	URL of the page to notify at the end of payment. Overrides the value entered in the notification settings.
vads_url_check_src	string (enum)	This parameter defines the source of the call of the notification URL (also called IPN URL).
vads_version	string	Version of the exchange protocol with the payment gateway.

Table 34: Parameter list - Technical details

Order details

Field name	Format	Description
vads_ext_info	ans	Customized field allowing to add optional data to the confirmation e-mail sent to the merchant.
vads_nb_products	n..12	Shopping cart details. Number of items.
vads_order_id	an..32	Order number.
vads_order_info	an..255	Order description.
vads_order_info2	an..255	Order description.
vads_order_info3	an..255	Order description.
vads_product_amountN	n..12	Shopping cart details. Amount of item N.

Field name	Format	Description
vads_product_labelN	an..255	Shopping cart details. Label of item N.
vads_product_qtyN	n..12	Shopping cart details. Quantity of item N.
vads_product_refN	an..64	Shopping cart details. Reference of item N.
vads_product_typeN	string (enum)	Shopping cart details. Type of item N.

Table 35: Parameter list - Order details

Buyer details

Field name	Format	Description
vads_cust_address	ans..255	Postal address.
vads_cust_address_number	an..5	Street number.
vads_cust_cell_phone	an..32	Cell phone number.
vads_cust_city	an..128	City.
vads_cust_country	a2	Country code in compliance with the ISO 3166 standard.
vads_cust_district	ans..127	District.
vads_cust_email	ans..150	Buyer's e-mail address.
vads_cust_first_name	an..63	First name.
vads_cust_id	an..63	Buyer reference on the merchant website.
vads_cust_last_name	an..63	Last name.
vads_cust_legal_name	an..100	Buyer's legal name.
vads_cust_name	an..127	Deprecated. Use **vads_cust_first_name** and **vads_cust_last_name**.
vads_cust_national_id	ans..255	National identifier.
vads_cust_phone	an..32	Phone number.
vads_cust_state	ans..127	State / Region.
vads_cust_status	string (enum)	Status (private / company).
vads_cust_title	an..63	Buyer's title.
vads_cust_zip	an..64	Postal code.

Table 36: Field list - Buyer details

Shipping details

Field name	Format	Description
vads_ship_to_city	an..128	City.
vads_ship_to_country	a2	Country code in compliance with the ISO 3166 standard.
vads_ship_to_delivery_company_name	ans..127	Transporter's name.
vads_ship_to_district	ans..127	District.
vads_ship_to_first_name	ans..63	First name.
vads_ship_to_last_name	ans..63	Last name.
vads_ship_to_legal_name	an..100	Legal name of the shipping place.
vads_ship_to_name	ans..63	Buyer's last name.
vads_ship_to_phone_num	ans..32	Phone number.
vads_ship_to_speed	string (enum)	Shipping mode
vads_ship_to_state	ans..127	State / Region.
vads_ship_to_status	string (enum)	Status (private / company).
vads_ship_to_street	ans..255	Postal address.
vads_ship_to_street_number	an..5	Street number.
vads_ship_to_street2	ans..255	Second line of the address.
vads_ship_to_type	string (enum)	Shipping type.
vads_ship_to_user_info	ans..255	Buyer details (CPF / CNPJ legal identifier).

Field name	Format	Description
vads_ship_to_zip	an..64	Zip code.

Table 37: Parameter list - Shipping details

Payment method details

Field name	Format	Description
vads_bank_code	n5	Code associated with the issuing bank.
vads_bank_product	an..3	Product code of the payment card.
vads_birth_day	n..2	Date of birth of the cardholder.
vads_birth_month	n..2	Month of birth of the cardholder.
vads_birth_year	n4	Year of birth of the cardholder.
vads_card_brand	an..127	Type of payment card.
vads_card_country	ISO 3166	Country code of the payment card.
vads_card_number	n..36	Masked card number.
vads_cvv	n..4	3 or 4-digit card security code.
vads_expiry_month	n..2	Expiration month of the payment card.
vads_expiry_year	n4	Expiration year of the payment card.

Table 38: Parameter list - Payment method details

Transaction details

Field name	Format	Description
vads_amount	n..12	The amount of the transaction presented in the smallest unit of the currency (cent for Euro).
vads_auth_mode	string (enum)	Specifies the mode of the authorization request.
vads_auth_number	an..6	Authorization number returned by the bank server.
vads_auth_result	n2	Return code of the authorization request returned by the issuing bank.
vads_capture_delay	n..3	Delay (in days) before the payment is captured.
vads_change_rate	string	Exchange rate used to calculate the effective payment amount (multi-currency payment).
vads_contract_used	ans..250	Merchant ID used.

vads_contracts	NETWORK1=contract1;
	NETWORK2=contract2;
		Merchant ID to use for the payment.

vads_currency	n3	Code of the currency to use for the payment.
vads_effective_amount	n..12	The amount of the payment presented in the smallest unit of the currency (cent for Euro) used for the capture.
vads_effective_creation_date	n14	Date of transaction registration in UTC format (GMT+0, 24H) (YYYYMMDDHHMMSS).
vads_operation_type	string (enum)	Allows to differentiate a debit from a credit (refund).
vads_payment_cards	type1;type2	List of the payment methods proposed to the buyer.
vads_payment_certificate	an40	The value of this field is populated by the payment gateway if the authorization has been **successfully** completed.
vads_payment_config	string (enum)	Payment type: immediate or installment.
vads_payment_option_code	string	Allows to specify the code of the chosen option.
vads_payment_seq	json	Describes the split payment sequence.
vads_payment_src	string (enum)	Payment source.
vads_risk_analyzer_result	ans	Returns the result of the risk management process performed by an external system (ClearSale, CyberSource, etc.).
vads_risk_assessment_result	ans	Returns the result of the risk assessment performed by the payment gateway.

vads_risk_control	control1=result1;
	control2=result2;
		List of risk management processes.

vads_sequence_number	n	Sequence (installment) number of the transaction.

Field name	Format	Description
vads_trans_date	n14	Date and time in UTC format (GMT+0, 24H) (YYYYMMDDHHMMSS).
vads_trans_id	n6	Unique transaction ID.
vads_trans_status	string (enum)	Transaction status.
vads_trans_uuid	ans32	Unique transaction ID generated by the payment gateway.
vads_validation_mode	string (enum)	Transaction validation mode.
vads_warranty_result	string (enum)	Transfer of responsibility in case the payment has been accepted.

Table 39: Parameter list - Transaction details

3DS Authentication

Field name	Format	Description
vads_threeds_cavv	ans..28	Indicates cardholder authentication via the ACS. Its value is populated by 3DS authentication server (ACS) when the buyer has been correctly identified (vads_threeds_status equals "Y" or "A").
vads_threeds_cavvAlgorithm	n1	Algorithm used by the ACS to generate the value of the CAVV. Its value is populated by the 3DS authentication server (ACS) when the buyer has been correctly identified (vads_threeds_status equals "Y" or "A").
vads_threeds_eci	n..2	Indicates the e-commerce index. Its value is populated by the 3DS authentication server (ACS) when the buyer has been correctly identified (vads_threeds_status equals "Y" or "A").
vads_threeds_enrolled	a1	Indicates the enrollment status of the cardholder. Its value is populated by the VISA and MASTERCARD (DS) servers during 3D Secure authentication.
vads_threeds_exit_status	n..2	Indicates the final status of 3D Secure authentication. Its value is populated by the payment gateway.
vads_threeds_mpi	n1	Enables / disables 3DS authentication for an e-commerce payment.
vads_threeds_sign_valid	n1	Indicates the validity of the PARes message signature. Its value is populated by the payment gateway.
vads_threeds_status	a1	Indicates the authentication status of the cardholder. Its value is populated by the 3DS authentication server (ACS) during the 3D Secure authentication.
vads_threeds_xid	ans..28	Indicates the unique 3DS authentication reference. Its value is populated by the authentication server (ACS) during the 3D Secure authentication.

Table 40: Field list - 3DS authentication

Donation details

Field name	Format	Description
vads_ext_info_donation	n..12	The donation amount presented in the smallest unit of the currency (cents for Euro).
vads_ext_info_donation_contribution	n..12	The donation amount (cents) presented in the smallest unit of the currency (cents for Euro).
vads_ext_info_donation_merchant	n8	ID of the shop that performed the donation.
vads_ext_info_donation_recipient	n..20	HelloAsso ID of the donation recipient.
vads_ext_info_donation_recipient_name	string	Name of the organization that received the donation.
vads_risk_primery_warranty	string (enum)	Allows to override the risk management configuration **Verification of transfer of responsibility for primary transactions**.

Table 41: Parameter list - Donation details

Customizing the payment page

Field name	Format	Description
vads_available_languages	language1;language2;language3	Allows to specify the list of languages available on the payment page.
vads_language	a2	Defines the language of the payment page (ISO 639-1 standard).
vads_shop_name	ans..127	Allows to define the shop name as it appears in the payment confirmation e-mails.
vads_shop_url	ans..127	Allows to override the shop URL that appears on the payment page and in payment confirmation e-mails.
vads_theme_config	code1=value1;code2=value2	Allows to customize some elements on the payment page.

Table 42: Parameter list - Customizing the payment page

Redirection to the merchant website

Field name	Format	Description
vads_redirect_error_message	ans..255	Automatic redirection: Message displayed on the payment page prior to redirection after an accepted / a declined payment.
vads_redirect_error_timeout	n..3	Automatic redirection: Delay (in seconds) before automatic redirection to the merchant website after an accepted / a declined payment.
vads_redirect_success_message	ans..255	Automatic redirection: Specifies the message that will appear upon automatic redirection to the merchant website if the payment has been accepted.
vads_redirect_success_timeout	n..3	Automatic redirection: Allows to define a delay in seconds before an automatic redirection to the merchant website at the end of an accepted payment. Its value is between 0 and 600s.
vads_return_mode	GET/POST/ NONE	Allows to specify the data transmission mode to the URLs of return to the merchant website.
vads_url_cancel	ans..127	URL where the buyer will be redirected after having clicked on "Cancel and return to shop" before proceeding to payment.
vads_url_check	ans..127	URL of the page to notify at the end of payment. Overrides the value entered in the notification settings.
vads_url_check_src	string (enum)	This parameter defines the triggering event of the instant notification (also called IPN).
vads_url_error	ans..127	URL where the buyer will be redirected in case of an internal processing error.
vads_url_refused	ans..127	URL where the buyer will be redirected in case the transaction has been declined.
vads_url_return	ans..127	Default URL where the buyer will be redirected after having clicked on "Return to shop".
vads_url_success	ans..127	URL where the buyer will be redirected in case of a successful transaction.

Table 43: Parameter list - Redirection to the merchant website

Subscription details

Field name	Format	Description
vads_identifier_status	string	Mandate registration status.
vads_recurrence_number	n	Recurrence number of the subscription
vads_recurrence_status	string	Recurring payment status.
vads_sub_amount	n..12	Amount of each installment except the ones that will be eventually defined by the **vads_sub_init_amount_number**

Field name	Format	Description
vads_sub_currency	n3	Numeric code of the currency to be used for the subscription in compliance with the ISO 4217.
vads_sub_desc	string	Rule for recurring payments to apply according to the iCalendar RFC5545 specification.
vads_sub_effect_date	n8	Subscription start date.
vads_sub_init_amount	n..12	Amount of the first installments.
vads_sub_init_amount_number	n..3	Number of installments for which the amount **vads_sub_init_amount** should be applied.
vads_subscription	ans..50	Subscription ID.

Table 44: Parameter list - Subscription details

signature

Description

Mandatory parameter.

Allows to verify the integrity of the transmitted requests.

This value is calculated:

- by the merchant website during the payment request,
- by the payment gateway during the response.

Format

ans 40

Error code

00 - signature Appears if the value of this field is incorrect,

70-empty params if the field is absent or empty.

Frequent errors:

- The fields of the form have not been encoded in UTF-8.
- Incorrect MODE (test or production) or CERTIFICATE.
- Line break and/or carriage return are posted in the form.
- Quotation marks ["] posted in the form.
- The transmitted signature does not respect the rule of signature computation.

Category

Technical details.

vads_action_mode

Description

Mandatory parameter.

Acquisition mode of the credit card details.

Format

string (enum)

Error code

47

Possible values

INTERACTIVE: input of the credit card details.

SILENT: input of the credit card details (requires a commercial option provided by you bank).

Category

Technical details

vads_amount

Description

The amount of the transaction presented in the smallest unit of the currency (lcents for Euro).

Example: for a transaction of 10 euros and 28 cents, the value of the parameter is 1028.

The payment form will be rejected in the following cases:

- an amount equal to zero [vads_amount=0],
- a negative amount [vads_amount=-100],
- an amount with decimals or points |vads_amount=100.50],
- a form without the vads_amount field (amount absent).

A message notifying of a technical error will be associated with a **09** return code (**vads_extra_result**).

Format n..12

Error code 09

Category Transaction details.

vads_auth_mode

Description

Field returned in the response.

Specifies how the authorization request is made.

Format string (enum)

Possible values

FULL : corresponds to an authorization for the total amount of the transaction.

Value used for immediate payments if the period between the requested capture date and the current date is shorter than the authorization validity period.

MARK : corresponds to an authorization for 1 euro.

Value used for deferred payments if the period between the requested capture date and the current date is strictly greater than the authorization validity period.

Category Transaction details.

vads_auth_number

Description

Field returned in the response.

Authorization number returned by the bank server, if available (otherwise, empty).

Format an..6

Category Transaction details.

vads_auth_result

Description Field returned in the response.

Code of the authorization request returned by the issuing bank, if available.

Format n..3

Possible values

Value	Description	Grounds of fraud	Value	Description	Grounds of fraud
00	Approved or successfully processed transaction		**38**	Expired card	
02	Contact the card issuer		**41**	Lost card	YES
03	Invalid acceptor	YES	**43**	Stolen card	YES
04	Keep the card	YES	**51**	Insufficient balance or exceeded credit limit	
05	Do not honor	YES	**54**	Expired card	YES
07	Keep the card, special conditions	YES	**55**	Incorrect secret code	
08	Confirm after identification		**56**	Card absent from the file	YES
12	Invalid transaction	YES	**57**	Transaction not allowed to this cardholder	YES
13	Invalid amount	YES	**58**	Transaction not allowed to this cardholder	
14	Invalid cardholder number	YES	**59**	Suspected fraud	YES
15	Unknown issuer	YES	**60**	Card acceptor must contact the acquirer	
17	Canceled by the buyer		**61**	Withdrawal limit exceeded	
19	Retry later		**63**	Security rules unfulfilled	YES
20	Incorrect response (error on the domain server)		**68**	Response not received or received too late	
24	Unsupported file update		**75**	Number of attempts for entering the secret code has been exceeded	
25	Unable to locate the registered elements in the file		**76**	The cardholder is already blocked, the previous record has been saved	YES
26	Duplicate registration, the previous record has been replaced		**90**	Temporary shutdown	
27	File update edit error		**91**	Unable to reach the card issuer	
28	Denied access to file		**94**	Duplicate transaction	
29	Unable to update		**96**	System malfunction	
30	Format error		**97**	Overall monitoring timeout.	
31	Unknown acquirer company ID	YES	**98**	Server not available, new network route requested.	
33	Expired card	YES	**99**	Initiator domain incident	
34	Fraud suspected	YES			

AMEX return code

Value	Description
000	Approved
001	Approve with ID
002	Partial Approval (Prepaid Cards only)
100	Declined
101	Expired Card / Invalid Expiration Date
106	Exceeded PIN attempts
107	Please Call Issuer
109	Invalid merchant
110	Invalid amount
111	Invalid account / Invalid MICR (Travelers Cheque)
115	Requested function not supported
117	Invalid PIN
119	Cardmember not enrolled / not permitted
122	Invalid card security code (a.k.a., CID, 4DBC, 4CSC)
125	Invalid effective date
181	Format error
183	Invalid currency code
187	Deny — New card issued
189	Deny — Account canceled
200	Deny — Pick up card
900	Accepted - ATC Synchronization
909	System malfunction (cryptographic error)
912	Issuer not available

Table 45: Amex card return code

Category Transaction details

vads_available_languages

Description

Allows to specify the list of languages available on the payment page.

The elements of the list must be separated by a semi-colon « ; ».

The languages on the payment page are represented by flags.

Format language1;language2;language3

Error code 71

Possible valuess

Language	Value	Flag shown by default
German	de	x
English	en	x
Chinese	zh	x
Spanish	es	x
French	fr	x
Italian	it	x
Japanese	ja	x
Dutch	nl	x
Polish	pl	
Portuguese	pt	x
Russian	ru	x
Swedish	sv	x
Turkish	tr	x

E.g.: to show the flags for French and English, post vads_available_languages=fr;en

Category Customization of the payment page.

vads_bank_code

Description

Field returned in the response.

Code associated with the issuing bank.

Format n5

Category Payment method details.

vads_bank_product

Description

Field returned in the response.

Product code of the payment card.

Format an..3

Possible values

VISA	Désignation
A	Visa Traditional
B	Visa Traditional Rewards
C	Visa Signature
D	Visa Signature Preferred
E	Proprietary ATM
F	Visa Classic
G	Visa Business
G1	Visa Signature Business
G2	Reserved
G3	Visa Business Enhanced
H	Reserved
I	Visa Infinite
J	Reserved
J1	Reserved
J2	Reserved
J3	Visa Healthcare
J4	Reserved
K	Visa Corporate T&E
K1	Visa GSA Corporate T&E
L	Electron
N	Visa Platinium
N1	TBA
P	Visa Gold
Q	Private Label
Q1	Reserved
R	Proprietary
S	Visa Purchasing
S1	Visa Purchasing
S2	Visa Purchasing
S3	Visa Purchasing
S4	Government Services Loan
S5	Commercial Transport EBT
S6	Business Loan
S7	Visa Distribution
T	Reserved
U	Visa TravelMoney
V	Visa VPay
W	Reserved
X	Reserved
Y	Reserved
Z	Reserved

MASTERCARD	Designation
MPN	MASTERCARD PREPAID DEBIT STANDARD-INSURANCE
MPO	MASTERCARD PREPAID DEBIT STANDARD-OTHER
MPP	MASTERCARD PREPAID CARD
MPR	MASTERCARD PREPAID DEBIT STANDARD-TRAVEL
MPT	MASTERCARD PREPAID DEBIT STANDARD-TEEN
MPV	MASTERCARD PREPAID DEBIT STANDARD-VERNMENT
MPW	DEBIT MASTERCARD BUSINESS CARD PREPAID WORK B2B
MPX	MASTERCARD PREPAID DEBIT STANDARD-FLEX BENEFIT
MPY	MASTERCARD PREPAID DEB STANDARD-EMPLOYEE INCENTIVE
MRG	MASTERCARD PREPAID CARD
MRH	MASTERCARD UNKNOWN PRODUCT
MRW	PREPAID MASTERCARD BUSINESS CARD
MSG	PREPAID MAESTRO CONSUMER RELOADABLE CARD
MSI	MAESTRO CARD
MWB	WORLD MASTERCARD FOR BUSINESS CARD
MWE	WORLD ELITE MASTERCARD CARD
DLS	DEBIT MASTERCARD CARD-DELAYED DEBIT
MCB	MASTERCARD BUSINESSCARD CARD
MCC	MASTERCARD CREDIT CARD (MIXED BIN)
MVOIR	MASTERCARD FLEET CARD
MCG	LD MASTERCARD CARD
MCO	MASTERCARD CORPORATE CARD
MCP	MASTERCARD PURCHASING CARD
MCS	STANDARD MASTERCARD CARD
MCW	WORLD MASTERCARD CARD
MDG	LD DEBIT MASTERCARD CARD
MDH	WORLD DEBIT EMBOSSED MASTERCARD CARD
MDP	PLATINUM DEBIT MASTERCARD CARD
MDS	DEBIT MASTERCARD CARD
MIU	DEBIT MASTERCARD UNEMBOSSED
MNW	MASTERCARD WORLD CARD
MOC	MASTERCARD UNKNOWN PRODUCT
MPG	DEBIT MASTERCARD STANDARD PREPAID-GENERAL SPEND
MPL	PLATINUM MASTERCARD CARD
MPP	MASTERCARD PREPAID CARD
MRG	MASTERCARD PREPAID CARD
MRO	MASTERCARD REWARDS ONLY
MRW	PREPAID MASTERCARD BUSINESS CARD
MSB	MAESTRO SMALL BUSINESS CARD
MSI	MAESTRO CARD
MSO	MAESTRO PREPAID OTHER CARD
MSW	PREPAID MAESTRO CORPORATE CARD
OLS	MAESTRO-DELAYED DEBIT
TCB	MASTERCARD BUSINESS CARD-IMMEDIATE DEBIT
TCC	MASTERCARD (MIXED BIN)-IMMEDIATE DEBIT
TCG	LD MASTERCARD CARD-IMMEDIATE DEBIT
TCS	MASTERCARD STANDARD CARD-IMMEDIATE DEBIT
TCW	WORLD SIGNIA MASTERCARD CARD-IMMEDIATE DEBIT
TNW	MASTERCARD NEW WORLD-IMMEDIATE DEBIT
TPL	PLATINUM MASTERCARD-IMMEDIATE DEBIT
WBE	MASTERCARD UNKNOWN PRODUCT

CB	Designation
1	National debit card
2	National credit and debit card
3	National credit card
4	National credit and debit card that requires systematic authorization
5	National credit card that requires systematic autorization

Category Payment method details.

vads_birth_day

Description Day of birth of the cardholder.

Format n..2

Error code 76

Category Payment method details.

vads_birth_month

Description Month of birth of the cardholder.

Format n..2

Error code 76

Category Payment method details.

vads_birth_year

Description Year of birth of the cardholder.

Format n4

Error code 78

Category Payment method details.

vads_capture_delay

Description Indicates the delay (in days) before the capture.

If the parameter is not submitted, the default value specified in the Back Office will be used. The default value can be configured in the Back Office by all authorized persons.

Format n..3

Error code 06

Category Transaction details.

vads_card_brand

Description

Field returned in the response.

Payment method used, if available (empty otherwise).

The value is derived from the BIN ranges files.

Format an..127

Possible values See the **vads_payment_cards** field.

Category Payment method details.

vads_card_country

Description

Field returned in the response.

Country code of the card in compliance with the ISO 3166 standard.

Format ISO 3166

Category Payment method details.

vads_card_number

Description

In the payment request

Card number (in case of a silent payment).

In the response

- Masked card number. Contains the 6 first digits of the number followed by "XXXXXX" and the 4 last numbers in the end.
- IBAN and BIC used for the payment separated by "_" in case of a direct debit payment.

Format an..36

Error code 40

Category Payment method details.

vads_change_rate

Description

Field returned in the response.

Exchange rate used to calculate the effective payment amount (multi-currency payment).

Format string

Category Transaction details

vads_contracts

Description	Presents a list with a Merchant ID (MID) to use for each acceptance network.
Format	NETWORK1=contract1;NETWORK2=contract2;NETWORK3=contract3
Error code	62
Possible values	The possible networks are:

Value	Description
ACCORD	Banque Accord network (private and gift cards)
ACCORD_SANDBOX	Banque Accord network (private and gift cards) - Sandbox mode
AMEX	American Express
ANCV	ANCV network (format: Convention #, Collector # and Reception # separated by "-" character). Eg.: vads_contracts=ANCV=123456-1-1
AURORE	Cetelem Aurore network (Brand cards and universal Aurore card)
CB	CB network (Visa, MasterCard, CB, e-CB, Maestro, Visa Electron)
CDGP	CDGP network (Privilege card)
CERIDIAN	Ceridian network (gift card)
COF3XCB	3xCB Cofinoga network
COF3XCB_SB	3xCB Cofinoga network - Sandbox mode
COFINOGA	Cofinoga network (Be Smart and brand cards)
GATECONEX	Gateconex network (Visa, MasterCard, CB, Maestro, Visa Electron, Diners)
GICC_MAESTRO	GICC network (Maestro card)
GICC_MASTERCARD	GICC network (MasterCard)
GICC_VISA	GICC network (VISA card)
GIROPAY	Giropay network
IDEAL	IDEAL network
JCB	JCB network
KLARNA	Klarna network
ONEY	Oney network
ONEY_SANDBOX	Oney network - Sandbox mode
PAYPAL	PayPal network
PAYPAL_SB	PayPal network - Sandbox mode
PAYSAFECARD	Paysafecard network
POSTFINANCE	Postfinance network
SEPA	SEPA network (DSS and SCT)
SOFORT	Sofort network
WIRECARD	WIRECARD network

For instance, if you have a second distant sale MID with a 12312312 number in your bank and you wish to register a credit card payment (Visa, MasterCard) for a specific order for this MID, you will need to populate **vads_contracts** as follows:

Example:

Note: this parameter is optional and is only used when **you have several e-commerce Merchant ID (MID)** within the same network and when you wish to **select a different Merchant ID (MID) depending on the payment.** If this parameter

	is not populated or absent, the payment will be registered for your default Merchant ID (MID).
Category	Information about the transaction.

vads_contract_used

Description	Field returned in the response. This field defines the value of the Merchant ID (MID) associated with the transaction. It is populated with the Merchant ID (MID) registered by default in your shop or it takes the value of the **vads_contracts** field sent in the payment request.
Format	ans..250
Category	Transaction details.

vads_contrib

Description	Optional information that indicates the name of the CMS used for the payment (Joomla, osCommerce, etc.). If you are developing yourself, this field can include the number of platform internal developer version.
Format	ans..128
Error code	31
Category	Technical details.

vads_ctx_mode

Description

Mandatory parameter.

Defines the context mode of the payment gateway.

Affects the choice of the certificate (test or production) to use for computing the signature.

The TEST certificate remains available even after the generation of the production certificate.

If you create a new merchant website (or have access to the acceptance testing environment), you can make tests without affecting the website that is currently in production.

Format string (enum)

Error code 11

Frequent errors:

- The mode has not been sent to the payment gateway.
- Do not use PROD for PRODUCTION in the code.
- Do not enter the value in lower case (test or production). This field only accepts unabbreviated values written in capital letters.

Possible values **TEST**, **PRODUCTION**

Category Technical details.

vads_currency

Description An ISO 4217 numerical code of the payment currency.

Format n3

Error code 10

Possible values

Currency	The ISO 4217 standard	Compatible with multi-currency option	Currency	The ISO 4217 standard	Compatible with multi-currency option
Thai Baht	764	✓	Moroccan Dirham	504	
Danish Crown	208	✓	Taiwan New Dollar	901	
Icelandic Crown	352		New Turkish Lira	949	✓
Norwegian Crown	578	✓	Kuwaiti Dinar	414	
Swedish Crown	752	✓	Tunisian Dinar	788	
Czech Crown	203	✓	Argentine Peso	032	
Australian Dollar	036	✓	Mexican Peso	484	
Canadian Dollar	124	✓	Philippine Peso	608	
Hong Kong Dollar	344	✓	South-African Rand	710	✓
Singapore Dollar	702	✓	Brazilian Real	986	✓
US Dollar	840	✓	Chinese Yuan (Renminbi)	156	✓
New Zealand Dollar	554	✓	Malaysian Ringgit	458	
Euro	978	✓	Russian Ruble	643	
Hungarian Forint	348	✓	Indonesian Rupiah	360	
CFP Franc	953	✓	South Korean Won	410	
Swiss Franc	756	✓	Japanese Yen	392	✓
Indian Rupee	356	✓	Polish Zloty	985	✓
Pound Sterling	826	✓			

Category Transaction details

vads_cust_address

Description Buyer's postal address.

Format ans..255

Note: *> and < >and <are not authorized.*

Error code 19

Note Mandatory parameter for 3xCB Cofinoga.

Category Buyer details.

vads_cust_address_number

Description Buyer's street number.

Format an..5

Error code 112

Category Buyer details.

vads_cust_cell_phone

Description Buyer's mobile phone number.

Format an..32

Error code 77

Category Buyer details.

vads_cust_city

Description Buyer's city.

Format an..128

Error code 21

Note Mandatory parameter for 3xCB Cofinoga.

Category Buyer details.

vads_cust_country

Description Buyer's country code in compliance with the ISO 3166 standard.

Format a2

Error code 22

Examples of possible values

Country	Code
BRAZIL	BR
CORSICA	FR
FRANCE	FR
GUADELOUPE	GP
MARTINIQUE	MQ
NEW CALEDONIA	NC
SAINT PIERRE AND MIQUELON	PM
POLYNESIE FRANCAISE	PF

Note Mandatory parameter for 3xCB Cofinoga.

Category Buyer details.

vads_cust_district

Description	Buyer's district.
Format	ans..127
Error code	113
Category	Buyer details.

vads_cust_email

Description	Buyer's e-mail address, **required if you want the buyer to receive e-mails from the payment gateway**. In order for the buyer to receive an e-mail, remember to post this parameter in the form when you generate a payment request.
Format	ans..150
Error code	15
Category	Buyer details.

vads_cust_first_name

Description	Buyer's first name.
Format	an..63
Error code	104
Note	Mandatory parameter for 3xCB Cofinoga.
Category	Buyer details.

vads_cust_id

Description	Buyer ID (identification by the merchant)
Format	an..63
Error code	16
Category	Buyer details.

vads_cust_last_name

Description	Buyer's last name.
Format	an..63
Error code	105
Note	Mandatory parameter for 3xCB Cofinoga.
Category	Buyer details.

vads_cust_legal_name

Description	Buyer's legal name.
Format	an..100
Error code	121
Category	Buyer details.

vads_cust_name

Description	Buyer's name. This field is deprecated. It is replaced by the fields **vads_cust_first_name** and **vads_cust_last_name**.
Format	an..127
Error code	18
Category	Buyer details.

vads_cust_national_id

Description	National identifier. Allows each citizen to identify him/herself with a unique ID within a country. For example, in Brazil, ClearSale requires this field to be populated with the CPF/CPNJ (in numeric format, between 11 and 20 digits long).
Format	ans..255
Category	Information about the buyer.

vads_cust_phone

Description	Buyer's phone number.
Format	an..32
Error code	23
Note	Mandatory parameter for 3xCB Cofinoga.
Category	Buyer details.

vads_cust_state

Description	Buyer's state/region.
Format	an..127
Error code	88
Category	Buyer details.

vads_cust_status

Description	Buyer type.
Format	string (enum)
Error code	92
Possible values	**PRIVATE**, **COMPANY**
Category	Buyer details.

vads_cust_title

Description	Buyer's marital status (e.g. Mr, Mrs, Ms).
Format	an..63
Error code	17
Possible values	**Mr**, **Mrs**, **Ms**
Note	Mandatory parameter for 3xCB Cofinoga.
Category	Buyer details.

vads_cust_zip

Description	Buyer's ZIP code.
Format	an..64
Error code	20
Note	Mandatory parameter for 3xCB Cofinoga.
Category	Buyer details.

vads_cvv

Description	In the payment request The card security code (in case of a silent payment). In the response Masked security number. Its length can vary between 3 and 4 digits depending on the card type.
Format	n..4
Error code	43
Category	Payment method details.

vads_effective_amount

Description	Field returned in the response. Payment amount in the effective currency used for the capture by the bank.
Format	n..12
Examples	**EXAMPLE FOR A SHOP WITH CAPTURE IN EUROS**

Payment of 10-Euro

Parameters sent in the payment form
vads_amount = 1000
vads_currency = 978

Returned parameters
vads_amount = 1000
vads_currency = 978
vads_effective_amount = 1000

Payment of 10 US Dollars

Parameters sent in the payment form
vads_amount = 1000
vads_currency = 840

Returned parameters
vads_amount = 1000
vads_currency = 840
vads_change_rate= 1.3118 (exchange rate)
vads_effective_amount = 762 (vads_amount / vads_change_rate)

An installment payment of 90 euros in 3 installments

Parameters sent in the payment form
vads_amount = 9000
vads_currency = 978
vads_payment_config=MULTI_EXT:date1=3000;date2=2000;date3=4000

Returned parameters for the first installment
vads_amount = 9000
vads_currency = 978
vads_effective_amount = 3000

An installment payment of 90-US Dollar paid in 3 installments

Parameters sent in the payment form

- vads_amount = 9000
- vads_currency = 840
- **vads_payment_config=MULTI_EXT:20121025=3000;20121026=2000;20121027=4000**

Returned parameters for the first installment

- vads_amount = 9000
- vads_currency = 840
- vads_change_rate= 1.3118 (exchange rate)
- vads_effective_amount = 2287 (**amount of the 1st installment, 30$ / vads_change_rate**)

Category Transaction details.

vads_effective_creation_date

Description Field returned in the response.

The transaction registration date, in UTC format (or GMT+0, 24H) (YYYYMMDDHHMMSS).

Format n14

Category Transaction details.

vads_expiry_month

Description	Expiration month of the payment card.
Format	n..2
Error code	41
Category	Payment method details.

vads_expiry_year

Description	Expiration year of the payment card.
Format	n4
Error code	42
Category	Payment method details.

vads_ext_info

Description	Allows to add an optional field to the confirmation e-mail sent to the merchant. It can be viewed in the Back Office, in the transaction details section (**Extras** tab). Its name must start with **vads_ext_info** to be taken into account. vads_ext_info_*fieldname*=value
Format	ans
Error code	91
Category	Order details.

vads_ext_info_bil_address_complement

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify additional information about the billing address.
Format	ans..250
Catégorie	Buyer details.

vads_ext_info_deadline

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify the delivery delay in days (N days).
Format	n
Category	Shipping details.

vads_ext_info_bil_date_of_birth

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify the buyer's date of birth on the receipt. Format : yyyy-mm-ddThh:mm:ss
Format	Datetime
Category	Buyer details.

vads_ext_info_bil_gender

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify on the receipt whether the buyer is male or female.
Format	n1
Category	Buyer details.

vads_ext_info_fingerprint_id

Description

Specific to Brazil and to the ClearSale fraud analyzer.

Unique session identifier.

- This identifier can be generated by the payment gateway.

  In this case, this parameter must not be populated.

- The identifier may also be generated by the merchant website

  In this case, this parameter must be populated with the desired value of the identifier. The merchant website must make sure that each identifier is unique. Any registration request containing an existing identifier will be rejected and will provoke an error message.

Format

string

It is encoded as 128 bytes and can contain uppercase or lowercase characters, numbers or hyphens ([A-Z] [a-z], 0-9, _, -).

Category

Buyer details.

vads_ext_info_ship_address_complement

Description	Specific to Brazil and to the ClearSale fraud analyzer. Allows to specify additional information about the shipping address.
Format	ans..250
Category	Shipping details.

vads_ext_info_ship_date_of_birth

Description

Specific to Brazil and to the ClearSale fraud analyzer.

Allows to specify the buyer's date of birth for the shipping.

Format Datetime

Category Shipping details.

vads_ext_info_ship_gender

Description

Specific to Brazil and to the ClearSale fraud analyzer.

Allows to specify for the shipping whether the buyer is male or female.

Format n1

Category Shipping details.

vads_ext_info_donation

Description

Parameter returned only in the instant notification (also called IPN) in case of a donation.

The donation amount presented in the smallest unit of the currency (cent for Euro).

Format n..12

Category Donation details.

vads_ext_info_donation_recipient

Description

Parameter returned only in the Instant Payment Notification for donations.

HelloAsso ID of the donation beneficiary.

Format n..20

Category Donation details.

vads_ext_info_donation_recipient_name

Description Parameter returned only in the instant notification (IPN) in case of a donation.

Name of the organization that received the donation.

Format string

Category Donation details.

vads_ext_info_donation_merchant

Description Parameter returned only in the instant notification (IPN) in case of a donation.

ID of the shop that performed the donation.

Format n8

Category Donation details.

vads_ext_info_donation_contribution

Description Parameter returned only in the instant notification (also called IPN) in case of a donation.

The donation amount (cents) presented in the smallest unit of the currency (cents for Euro).

Format n..12

Category Donation details.

vads_extra_result

Description Field returned in the response.

Optional code of the response. Its meaning depends on the value entered in **vads_result**.

- If **vads_result** equals 30 (request error), then **vads_extra_result** contains the numerical code of the field with an error in the value or the format. This value can be set to 99 in case of an unknown error in the request.

  Example: if **vads_extra_result** contains the value 09, it means that the amount specified in **vads_amount** is incorrect (for example, if the amount contains decimals, as it would not have been converted to cents in advance).

- If **vads_result** equals 05 (declined) or 00 (accepted), **vads_extra_result** contains the numerical code of the risk management result.

vads_extra_result	Description
Empty	No verification completed.
00	All the verification processes have been successfully completed.
02	Credit card velocity exceeded.
03	The card is in the merchant's greylist.
04	The country of origin of the card is in the merchant's greylist.
05	The IP address is in the merchant's greylist.
06	The BIN code is in the merchant's greylist.
07	Detection of an e-carte bleue.
08	Detection of a national commercial card.
09	Detection of a foreign commercial card.
14	Detection of a card that requires systematic authorization.
20	Relevance verification: countries do not match (country IP address, card country, buyer's country).
30	The country of the IP address is in the greylist.
99	Technical issue encountered by the server during a local verification process.

Category Technical details.

vads_hash

Description A unique key sent only to the IPN.

Format an64

Category Technical information.

vads_identifier

Description A unique identifier (token or unique mandate reference) associated with a payment method.

- This identifier can be generated by the payment gateway.

  In this case, this parameter must not be populated.

- It can also be generated by the merchant website.

  In this case, this parameter must be populated with the desired value of the identifier. **The merchant website must make sure that each identifier is unique**. Any registration request containing an existing identifier will be rejected and will provoke an error message.

Format ans..50

Error code 30

Category Subscription details.

vads_identifier_status

Description

Field returned in the response.

Appears only if the requested action concerns creating or updating:

- a token (subscription)
- a UMR (SEPA Unique Mandate Reference)

Format

string

Possible values

Value	Description
CREATED	The authorization request has been accepted. Token or UMR has been successfully created.
NOT_CREATED	The authorization request has been declined. The token or UMR has not been created and will not be displayed in the Back Office.
UPDATED	The token or UMR has been successfully updated.
NOT_UPDATED	The token or UMR has not been updated.
ABANDONED	The action has been abandoned by the buyer (debtor). The token or UMR has not been created and cannot be viewed in the Back Office.

Category

Subscription details.

vads_insurance_amount

Description

Amount of insurance fees for the entire order.

Concerns only the PayPal payment method.

Format

n..12

Error code

110

Category

Order details.

vads_language

Description

In the payment request:

Defines the language of the payment page (ISO 639-1 standard).

If the field has not been sent in the payment request or is empty, the payment page will be shown in the language of the buyer's browser.

In the response:

Returns the value specified in the form if the buyer has not changed the language of the payment page.

Returns the language selected by the buyer if the buyer has changed it by clicking on a different flag..

Format

a2

Error code

12

Possible values

Language	ISO 639-1 standard
German	de
English	en
Chinese	zh
Spanish	es
French	fr
Italian	it
Japanese	ja
Dutch	nl
Polish	pl
Portuguese	pt
Russian	ru
Swedish	sv
Turkish	tr

Category Customization of the payment page

vads_nb_products

Description Allows to define the number of items in the cart.

Format n..12

Category Order details.

vads_operation_type

Description Field returned in the response.

Allows to differentiate a debit from a credit (refund).

Note:

vads_operation_type *is not returned in the response when a payment is canceled or abandoned*

Format string (enum)

Possible values **DEBIT**, **CREDIT**

Category Transaction details.

vads_order_id

Description Order ID. It is also included in the payment confirmation e-mail sent to the buyer.

Alphanumeric field. Only one special character, "-", is allowed.

If any other special characters are used (&, ;, @, etc.), the payment gateway will return an error.

Format an..32

Error code 13

Category Order details.

vads_order_info

Description	Order description.
Format	an..255
Error code	14
Category	Order details.

vads_oder_info2

Description	Order description.
Format	an..255
Error code	14
Category	Order details.

vads_order_info3

Description	Order description.
Format	an..255
Error code	14
Category	Order details.

vads_page_action

Description	Mandatory parameter. Defines the action that must be performed.
Format	string (enum)
Error code	46
Possible values	• PAYMENT • REGISTER • REGISTER_UPDATE • REGISTER_PAY • REGISTER_SUBSCRIBE • REGISTER_PAY_SUBSCRIBE • SUBSCRIBE • REGISTER_UPDATE_PAY • ASK_REGISTER_PAY
Category	Technical details.

vads_payment_cards

Description

Contains the list of card types proposed to the buyer, separated by a "**;**".

If this list contains only one card type, the data entry page for this payment method will directly appear. If there is more than one card type, the payment method selection page will appear.

If this parameter is empty (recommended), the available payment methods (currencies, technical constraints, etc.) associated with the shop will be proposed.

Format

type1;type2;type3

Error code

08

The form will be rejected when:

- the transmitted value is not in the following list of values.
- the values: TOUTES, ALL are not accepted. To provide all the payment methods, this parameter should not been posted or be posted empty,
- the transmitted value does not correspond to the payment method available for your shop,
- Your e-commerce Merchant ID (MID) was closed by your bank. Contact the payment gateway customer service to fix the problem.
- the transmitted value is not value is not eligible for the associated network.

Possible values

Value	Description
ACCORD_STORE	Banque Accord card
ACCORD_STORE_SB	Banque Accord card - Sandbox
ALINEA	Alinea Private Card
ALINEA_CDX	Alinea Gift Card
ALINEA_CDX_SB	Alinea Gift Card - Sandbox
ALINEA_SB	Alinea Private Card - Sandbox
AMEX	American Express
AURORE-MULTI	Aurore Card
BANCONTACT	Maestro Bancontact Mistercash Card
BITCOIN	Payment by virtual currency
BIZZBEE_CDX	Bizzbee Gift Card
BIZZBEE_CDX_SB	Bizzbee Gift Card - Sandbox
BRICE_CDX	Brice Gift Card
BRICE_CDX_SB	Brice Gift Card - Sandbox
CB	CB
CDGP	Privilege Card
COF3XCB	3xCB Cofinoga
COF3XCB_SB	3xCB Cofinoga - Sandbox
COFINOGA	Be Smart Card
CORA_BLANCHE	Cora Blanche Card
CORA_PREM	Cora Premium Card
CORA_VISA	Cora Visa Card
DINERS	Diners Club Card
E-CARTEBLEUE	E-carte bleue (French virtual card)
E_CV	E-chèque vacance
EDENRED_TR	"Ticket Restaurant"
EDENRED_EC	"Ticket EcoChèque"

Value	Description
GIROPAY	Wire transfer
KLARNA	Invoice payment
IDEAL	Wire transfer
ILLICADO	Illicado Gift Card
ILLICADO_SB	Illicado Gift Card - Sandbox
JCB	JCB Card
JOUECLUB_CDX	Jouéclub Gift Card
JOUECLUB_CDX_SB	Jouéclub Gift Card - Sandbox
JULES_CDX	Jules Gift Card
JULES_CDX_SB	Jules Gift Card - Sandbox
MAESTRO	Maestro
MASTERCARD	MasterCard
ONEY	Payment in 3/4 installments Oney FacilyPay
ONEY_SANDBOX	Payment in 3/4 installments Oney FacilyPay - Sandbox
PAYLIB	Paylib
PAYPAL	PayPal
PAYPAL_SB	PayPal - Sandbox
PAYSAFECARD	Paysafecard Prepaid Card
POSTFINANCE	PostFinance
POSTFINANCE_EFIN	E-finance Postfinance mode
RUPAY	Indian Payment method
SCT	SEPA Credit Transfer
SDD	SEPA Direct Debit
SOFORT_BANKING	Sofort
TRUFFAUT_CDX	Truffaut Gift Card
VISA	Visa
VISA_ELECTRON	Visa Electron

Category Transaction details

vads_payment_certificate

Description Field returned in the response.

This field is populated by the payment gateway if the authorization has been **successfully** completed.

Format an40

Category Transaction details.

vads_payment_config

Description

Defines the type of payment: immediate or installment.

- For a single payment, the value must be set to **SINGLE**.
- For an installment payment with fixed amounts and dates, the value must be set to **MULTI:** followed by key=value pairs separated by the "**;**" character.

  The parameters are:

  - « **first** » indicates the amount of the first installment (populated in the smallest unit of the currency).
  - « **count** » indicates the total number of installments.
  - "**period**" indicates the number of days between 2 installments.

    The field order associated with MULTI must be respected.
- For an installment payment with a customized installment schedule, the value must be set to **MULTI_EXT:** followed by the date=amount pairs separated by the "**;**" character.

  The dates must not be passed.

The MULTI_EXT value requires a subscription to the **Advanced installment payment** option.

Format

string (enum)

Error code

07

Possible values

- **SINGLE**
- **MULTI:first= initial_amount ;count= installments_nb ;period=interval_in_days**
- **MULTI_EXT:date1=amount1;date2=amount2;date3=amount3**

Example 1

MULTI allows to define an installment payment.

The amount of each installment corresponds to the total amount divided by the number of installments.

The amount of the first installment can be different, it can be specified in **first**.parameter.

In case the remaining amount does not equal zero, it will be added up to the amount of the last installment.

Payment request:

- vads_capture_delay=2
- vads_currency=978
- vads_amount=20000
- vads_payment_config=MULTI:first=10000;count=4;period=30

Result:

A first payment of 100 euros will be captured by the bank in 2 days (vads_capture_delay).

A second payment of 33,33 euros will be made in 32 days vads_capture_delay + period.

A third payment of 33,33 euros will be made in 62 days.

A fourth payment of 33,34 euros will be made in 92 days.

The total amount is 200 euros (vads_amount= 20000). The remaining amount has been added to the amount of the last installment.

This instruction allows to immediately create 4 payments with the same transaction number but different sequence numbers (vads_sequence_number).

Example 2

MULTI_EXT allows to define a customized installment schedule. You will be able to define the amount of each installment.

Payment request MULTI_EXT :

- vads_currency=978
- vads_amount=10000
- vads_payment_config= MULTI_EXT**:**20150601 =5000; 20150701 =2500; 20150808 =2500

Result:

The first payment of 50 € is scheduled for June 1st 2015.

The second payment of 25 € is scheduled for July 1st 2015.

The third payment of 25 € is scheduled for August 8th 2015.

Note:

The total amount must be equal to the value of the **vads_amount** field. The date of the last installment cannot be later than 12 months after the date of submission of the form. If the last installment is scheduled later than the card expiry date, no installment will be registered and the buyer will be notified about this issue.

Category

Transaction details.

vads_payment_error

Description

Field returned in the response.

Error codes that may appear when a payment has been declined.

Format n..3

Possible values

Error code	Error message	Error code	Error message
1	Transaction not found.	72	The authorization has been declined by Cofinoga.
2	Transaction not found.	73	The authorization for 1 euro has been declined.
3	This action has not been authorized for a transaction with the {0} status.	74	Invalid payment configuration.
4	This transaction is not authorized in this context.	75	The operation has been rejected by PayPal.
5	This transaction already exists.	76	Due to a technical problem, we are unable to process your request.
6	Invalid transaction amount.	77	Due to a technical problem, we are unable to process your request.
7	This action is not possible anymore for a transaction created on that day.	78	Transaction ID missing.
8	The card expiration date does not allow this action.	79	This transaction ID is already used.
9	CVV mandatory for this card.	80	Transaction ID expired.
10	The refund amount is greater than the initial amount.	81	Invalid contents of the configuration theme.
11	The refunds total amount is greater than the initial amount.	82	The refund is not authorized.
12	Credit duplication (refund) is not authorized.	83	The transaction amount does not respect the allowed values.
13	Due to a technical problem, we are unable to process your request.	84	Due to a technical problem, we are unable to process your request.
14	Due to a technical problem, we are unable to process your request.	85	Due to a technical problem, we are unable to process your request.
15	Due to a technical problem, we are unable to process your request.	86	Due to a technical problem, we are unable to process your request.
16	Due to a technical problem, we are unable to process your request.	87	Due to a technical problem, we are unable to process your request.
17	Aurore Merchant ID (MID) configuration has failed.	88	Due to a technical problem, we are unable to process your request.
18	Cetelem response analysis has failed.	89	The modification is not authorized.
19	Unknown currency.	90	An error occurred during refund.
20	Invalid card type.	91	No payment options have been enabled for this Merchant ID (MID).
21	No Merchant ID (MID) found for this payment. Please modify the data or contact your manager in case the error reoccurs.	92	An error occurred while calculating the payment channel.
22	Shop not found.	93	An error occurred during buyer redirection to the page of payment finalization.
23	Ambiguous Merchant ID (MID).	94	A technical error has occurred.
24	Invalid Merchant ID (MID).	95	Due to a technical problem, we are unable to process your request.
25	Due to a technical problem, we are unable to process your request.	96	An error occurred at the moment of capture of this transaction.
26	Invalid card number.	97	The capture date is too late.
27	Invalid card number.	98	Invalid transaction date.

Error code	Error message	Error code	Error message
28	Invalid card number.	99	An error occurred while calculating the payment source.
29	Invalid card number.	100	Failed commercial card verification.
30	Invalid card number (Luhn).	101	Rejected due to the refusal of the first installment.
31	Invalid card number (length).	102	The operation has been declined by Buyster.
32	Invalid card number (not found).	103	The transaction status could not be synchronized with the external system.
33	Invalid card number (not found).	104	An error occurred at the moment of capture of this transaction.
34	Failed verification of the card requiring systematic verification.	105	A security error occurred while processing 3DS authorization for this transaction.
35	Failed e-Carte Bleue verification.	106	Unsupported currency for this Merchant ID (MID) and/or shop.
36	The transaction has been refused by risk management.	107	The card associated with the token is not valid anymore.
37	Interruption not processed during the payment.	108	Due to a technical problem, we are unable to process your request.
38	Due to a technical problem, we are unable to process your request.	109	The timeout has been exceeded during buyer redirection.
39	3D Secure refusal for the transaction.	110	Payment card not supported by the Merchant ID (MID).
40	Due to a technical problem, we are unable to process your request.	111	The transactions have been declined without liability shift.
41	Due to a technical problem, we are unable to process your request.	112	Cancellation is not authorized.
42	An internal error occurred while consulting the card number.	113	Duplication is not authorized.
43	An internal error occurred while consulting the card number.	114	Override is not authorized.
44	Unable to force an authorization for 1 euro.	115	The refund is not authorized.
45	Invalid currency for the modification.	116	Manual payment not authorized for this card.
46	The amount is greater than the authorized amount.	118	Manual installment payment not authorized for this card.
47	The desired capture date exceeds the authorization expiration date.	119	The submitted date is invalid.
48	The requested modification is invalid.	120	The initial transaction option is not applicable.
49	Invalid definition of the installment payment.	124	Inactive card.
50	Unknown shop.	125	Payment refused by the acquirer.
51	Unknown exchange rate.	126	This action is impossible as the payment sequence has not been completed.
52	The Merchant ID (MID) has been terminated since {0}.	132	Due to a technical problem, we are unable to process your request.
53	The shop {0} has been closed since {1}.	135	Integration of a payment page into an iframe is not authorized.
54	Rejected parameter that may contain sensitive data {0}.	136	The derivative transactions have been declined without a transfer of responsibility for the initial transaction.
55	Due to a technical problem, we are unable to process your request.	137	Duplicate transaction.
57	An error occurred while retrieving the token.	138	Partial refund is impossible for this transaction.
58	The token status is not compatible with this operation.	139	Refund declined.
59	An error occurred while retrieving the token.	141	The transaction has been refused by the risk analyzer.
60	This token already exists.	142	The card type used is not valid for the requested payment mode.

Error code	Error message	Error code	Error message
61	Invalid token.	143	Due to a technical problem, we are unable to process your request.
62	Token creation failed.	144	A transaction in production mode has been marked as in test mode by the acquirer.
63	This subscription already exists.	145	A transaction in test mode has been marked as in production mode by the acquirer.
64	This subscription is already terminated.	146	Invalid SMS code.
65	Invalid subscription.	147	The risk management module has requested for this transaction to be declined.
66	Invalid recurrence rule.	148	Due to a technical problem, we are unable to process your request. The transaction has not been created.
67	Subscription creation failed.	149	The payment session has expired (the buyer has been redirected to the ACS and has not finalized the 3D Secure authentication).
69	Due to a technical problem, we are unable to process your request.	150	Due to a technical problem, we are unable to process your request. The transaction has not been created.
70	Invalid country code.		
71	Invalid web service parameter.		

Category Technical details.

vads_payment_option_code

Description Allows to specify the code of the chosen option.

Format string

Error code 103

Category Transaction details.

vads_payment_seq

Description Details of completed transactions.

Format json

vads_payment_seq (json format) describes the split payment sequence. It contains:

- **"trans_id"**: the global transaction identifier to the payment sequence.
- **"transaction"**: the transaction table of the sequence. It contains:

Field name	Description
amount	Amount of the payment sequence.
operation_type	Debit transaction.
auth_number	Authorization number returned by the bank server, if available (otherwise, empty). Note: this parameter is returned empty for payments by "e-Chèque-Vacances". It is not provided by the ANCV server.
capture_delay	Delay (in days) before the capture. • For a payment by card, the value of this parameter takes into account the delay (in days) before the capture in the bank. If the parameter is not transmitted in the payment form, the default value specified in the Back Office will be used. Note: for payments by "e-Chèques-Vacances", this parameter is set to **0**. The "e-Chèques-Vacances" are validated in real time.
card_brand	Payment method used. • For a payment by card, e.g. Visa card, this parameter is set to **CB**. • For payments by "e-Chèques-Vacances", this parameter is set to **E_CV**.
card_number	Payment method number. • For a payment by "e-Chèques-Vacances", this parameter is set by concatenating the number of the "e-Chèques-Vacances" with the organization number and the year of the issue separating them with a "-". • For a payment by card, the number is masked.
expiry_month	Expiration month of the payment card. Missing parameter for a payment by "e-Chèque-Vacances".
expiry_year	Expiration year of the payment card. Missing parameter for a payment by "e-Chèque-Vacances".
payment_certificate	Payment certificate.
presentation_date	• For a payment by card, this parameter is the requested capture date (ISO 8601 format). • For payments by "e-Chèques-Vacances", this parameter corresponds to the order date. The "e-Chèques-Vacances" are validated in real time by the ANCV server.
trans_id	Transaction number.
ext_trans_id	Missing parameter for a payment card. For a payment by "e-Chèques-Vacances", this parameter is populated by the payment gateway (transaction reference).
sequence_number	Sequence number.
trans_status	Transaction status.

Table 46: JSON object content

Note: canceled transactions also appear in the table (this information appears in the JSON trans_status field).

Category Transaction details.

vads_payment_src

Description Allows to define the payment source.

Format string (enum)

Error code 60

Possible values

Value	Description
EC	E-commerce: payment made on the payment page.
MOTO	MAIL OR TELEPHONE ORDER : payment processed by an operator following a MOTO order.
CC	Call center : payment made through a call center.
OTHER	Other: payment made through a different source, e.g. Back Office.

Category Transaction details.

vads_pays_ip

Description Field returned in the response.

Country code of the buyer's IP address in compliance with the ISO 3166 standard.

Format a2

Category Buyer details.

vads_presentation_date

Description Field returned in the response.

- Requested capture date.

  or

- Requested presentation date for a SEPA Direct Debit.

Format n14

Category Transaction details.

vads_product_amountN

Description Allows to define the amount of each item in the cart.

N corresponds to the item index. (0 for the first one, 1 for the second one, etc.).

The amount must be populated in the smallest unit of the currency. Cents for Euro.

Format n..12

Error code 102

Category Order details.

vads_product_labelN

Description Allows to define the name of each item in the cart.

N corresponds to the item index. (0 for the first one, 1 for the second one, etc.).

Format an..255

<u>Note:</u>

Alphabetical and numerical characters must be used (for example, space is not authorized). Please refer to the RegEx ***^[a-zA-Z0-9]{1,255}$*** *to define the name of each item in the cart.*

Error code 97

Category Information about the order.

vads_product_qtyN

Description Allows to define the quantity of each item in the cart.

N corresponds to the item index. (0 for the first one, 1 for the second one, etc.).

Format n..12

Error code 101

Category Order details.

vads_product_refN

Description Allows to define the reference of each item in the cart.

N corresponds to the item index. (0 for the first one, 1 for the second one, etc.).

Format an..64

Error code 100

Category Order details.

vads_product_typeN

Description

Allows to define the type of each item in the cart.

N corresponds to the index of the item. (0 for the first one, 1 for the second one, etc.).

Format

string (enum)

Error code

98

Possible values

Value	Description
FOOD_AND_GROCERY	Food and grocery
AUTOMOTIVE	Cars / Moto
ENTERTAINMENT	Entertainment / Culture
HOME_AND_GARDEN	Home and gardening
HOME_APPLIANCE	Household appliances
AUCTION_AND_GROUP_BUYING	Auctions and group purchasing
FLOWERS_AND_GIFTS|	Flowers and presents
COMPUTER_AND_SOFTWARE	Computers and software
HEALTH_AND_BEAUTY	Health and beauty
SERVICE_FOR_INDIVIDUAL	Services for individuals
SERVICE_FOR_BUSINESS	Services for companies
SPORTS	Sports
CLOTHING_AND_ACCESSORIES	Clothes and accessories
TRAVEL	Travel
HOME_AUDIO_PHOTO_VIDEO	Sound, image and video
TELEPHONY	Telephony

Table 47: Values associated with vads_product_typeN

Category

Transaction number

vads_product_vatN

Description

Allows to define the tax for each item in the cart.

N corresponds to the item index. (0 for the first one, 1 for the second one, etc.).

Format

n..12

Error code

203

Possible values

- **An integer without a decimal separator**

  To display an amount in cents applied to the product in question.

  Example in euros: 14520 (for an amount of 145 euros and 20 cents)

- **An integer less than 100 with a decimal separator**

  To display a percentage applied to the payment amount for the product in question with maximum 4 digits after the decimal point.

  Examples: 20.0 or 19.6532

  Notes:

  - *The decimal separator is mandatory for displaying a percentage*.
  - *The decimal separator is represented by the "***.***"* symbol.

Category Order details.

vads_recurrence_number

Description

Field returned in the response.

Recurrence number of the subscription.

Format n

Category Subscription details.

vads_recurrence_status

Description

Field returned in the response.

Subscription status.

Appears only if the requested action concerns creating or updating a subscription (REGISTER_SUBSCRIBE, SUBSCRIBE, REGISTER_PAY_SUBSCRIBE).

Format string

Possible values

Value	Description
CREATED	The subscription has been successfully created. Subscription details are visible in the Back Office.
NOT_CREATED	The subscription has not been created and is not visible in the Back Office.
ABANDONED	The request for creating a subscription has been abandoned by the buyer (debtor). The subscription has not been created and is not visible in the Back Office.

Category Subscription details.

vads_redirect_error_message

Description	Allows to define the message that will appear before automatic redirection to the merchant website if the payment has been declined.
Format	ans..255
Error code	37
Category	Redirection to the merchant website.

vads_redirect_error_timeout

Description	Allows to define a delay in seconds before an automatic redirection to the merchant website at the end of a declined payment. The value of the field is between **0** and **600**s. After this delay, the buyer will be redirected to the URL populated in the **vads_url_refusal** field. If it is not set, the buyer will be redirected to the Return URL populated in **vads_url_return** or to the Return URL specified in the Back Office. If the Return URL is not set, the buyer will be redirected to the shop URL.
Format	n..3
Error code	36
Category	Redirection to the merchant website.

vads_redirect_success_message

Description	Allows to define the message that will appear upon automatic redirection to the merchant website if the payment has been accepted.
Format	ans..255
Error code	35
Category	Redirection to the merchant website.

vads_redirect_success_timeout

Description	Allows to define a delay in seconds before an automatic redirection to the merchant website at the end of an accepted payment. Its value is between 0 and 600 sec. After this delay, the buyer will be redirected to the URL populated in the **vads_url_success** field. If it is not set, the buyer will be redirected to the Return URL populated in **vads_url_return** or to the Return URL specified in the Back Office.If the Return URL is not set, the buyer will be redirected to the shop URL.
Format	n..3
Error code	34
Category	Redirection to the merchant website.

vads_result

Description Field returned in the response.

Return code of the requested action.

Format n2

Possible values

Value	Description
00	Action successfully completed.
02	The merchant must contact the cardholder's bank. Deprecated.
05	Action rejected.
17	Action canceled by the buyer.
30	Request format error. To match with the value of the vads_extra_result field.
96	Technical issue.

Category Technical details.

vads_return_mode

Description Allows to specify the data transmission method used while returning to the merchant website.

Format string (enum)

Error code 48

Possible values

Field name	Value	Description
vads_return_mode	absent, empty or **NONE**	No parameters will be transmitted to the Return URL.
	GET	The return fields will be transmitted to the return URL in an HTTP GET form (in the "query string").
	POST	The return fields will be transmitted to the return URL in an HTTP POST form. If the return to the shop in done from an environment **other than https**, a security pop-up message will be displayed to the buyer.

Category Redirection to the merchant website.

vads_risk_analyzis_result

Description Field returned in the response.

Returns the result of the risk management process performed by an external system (ClearSale, CyberSource, etc.).

Returned in the instant notification URL (also called IPN) and in return parameters.

Format ans

Possible values

Values common to all types of risks analyzers	
INVALID_CREDENCIAL	Configuration problem of the risk analyzer contract.
COMUNICATION_PROBLEM	Miscommunication.
DATA_PROCESSING_PROBLEM	Problem while processing the request data.
MISSING_MANDATORY_ORDER_INFO	Order information missing.
MISSING_MANDATORY_SHIPPING_INFO	Shipping information missing.
MISSING_MANDATORY_SHIPPING_ADDRESS_INFO	Shipping address information missing.
MISSING_MANDATORY_BILLING_INFO	Billing information missing.
MISSING_MANDATORY_BILLING_ADDRESS_INFO	Billing information missing.
MISSING_MANDATORY_CARD_INFO	Payment method information missing.
MISSING_MANDATORY_CUSTOMER_INFO	Buyer information missing.

Table 48: Values associated with vads_risk_analyzis_result common to all types of risks analyzers

ClearSale		
APA	**Automatically approved**	The transaction is automatically approved according to the defined parameters.
APM	**Manually approved - order manually approved by analyst's decision**	The transaction is manually approved by an analyst.
RPM	**Reproved with no suspect**	The order is reproved due to missing information related to the buyer in conformity with the policy in force.
AMA	**Waiting for manual analysis - order is in a queue waiting for analysis**	Waiting for manual analysis. The order is waiting to be analyzed.
ERR	**Error**	Error
NVO	**New order - order waiting for score**	New order. Waiting to be processed and classified.
SUS	**Suspended order - order suspended by fraud suspicion**	Order manually suspended. The order is suspended for suspected fraud.
CAN	**Canceled - order canceled by user**	Order canceled. The order has been canceled by the buyer.
FRD	**Order confirmed as a fraud**	Fraud confirmed by the credit card operator or the cardholder.
RPA	**Automatically reproved based on parameters within risk analyzer**	Order automatically reproved. The order is reproved based on the parameters of the external risk analyzer.
RPP	**Automatically reproved based customer or ClearSale policy**	Order automatically reproved. The order is reproved based on the customer or ClearSale policy.

Table 49: Values associated with vads_risk_analyzis_result - ClearSale

CyberSource		
100	**SUCCESS**	The transaction is successfully completed.
101	**MISSING_FIELDS**	The transaction has been declined. One or more parameters are missing.
102	**INVALID_FIELDS**	The transaction has been declined. One or more parameters contain invalid data.
150	**ERROR_GENERAL_SYSTEM_FAILURE**	Error.
151	**SERVER_TIME_OUT**	Error. The request was received but the time limit has been exceeded. This error does not include timeouts between the client and the server.
152	**SERVICE_TIME_OUT**	Error. The request was received but a service was not completed in time.

CyberSource		
202	**CARD_EXPIRED**	Declined. Card expired.
231	**ACCOUNT_NUMBER_INVALID**	Declined. Invalid card number.
234	**ACCOUNT_PROBLEM**	Declined. A problem occurred with the merchant CyberSource configuration.
400	**FRAUD_SCORE_TOO_HIGH**	Declined. The score of the fraud exceeds the tolerance.
480	**SUCCESS_TO_REVIEW**	The order is marked and needs to be reviewed by the Decision Manager.
481	**SUCCESS_TO_REJECT**	The order has been declined by Decision Manager.

Table 50: Values associated with vads_risk_analyzis_result - Cybersource

Category Transaction details.

vads_risk_assessment_result

Description Field returned in the response.

Returns the result of the advanced risk assessment performed by the payment gateway.

Returned in the IPN URL and in the return parameters.

Format ans

Possible values

Values	Description
ENABLE_3DS	3D Secure enabled
DISABLE_3DS	3D Secure disabled
MANUAL_VALIDATION	The transaction has been created via manual validation. The payment capture is temporarily blocked to allow the merchant perform all the desired verifications.
REFUSE	The transaction is refused.
RUN_RISK_ANALYSIS	Call for an external risk analyser if the merchant has a contract. Refer to the description of the **vads_risk_analysis_result** field to identify the list of possible values and their description.
INFORM	A warning message appears. The merchant is notified that a potential problem has been identified. The merchant is informed via one or several notification centre rules (IPN, e-mail or SMS).

Category Transaction details.

vads_risk_control

Description

Field returned in the response.

Allows to define the outcome of the risk management process.

Format

control1=result1;control2=result2

Possible values

Value	Description
CARD_FRAUD	Verifies if the cardholder's card number is in the card greylist.
SUSPECT_COUNTRY	Verifies if the cardholder's card country is in the forbidden countries list.
IP_FRAUD	Verifies if the cardholder's IP address is in the IP greylist.
CREDIT_LIMIT	Verifies if frequency and the amounts of the purchases made with the same card number, or the maximum order amount.
BIN_FRAUD	Verifies if the card BIN code is in the BIN code greylist.
ECB	Verifies if the buyer's card is an e-carte bleue.
CARD_COMMERCIAL	Verifies if the buyer's card is a commercial card.
SYSTEMATIC_AUTO	Verifies if the buyer's card is a card that requires systematic authorization.
INCONSISTENT_COUNTRIES	Verifies if the country of the IP address, the country of the payment card and the country of residence of the buyer match.
NON_WARRANTY_PAYMENT	Verifies the transfer of responsibility of the transaction.
SUSPECT_IP_COUNTRY	Verifies that the cardholder's country, identified by his/her IP address, is in the forbidden countries list.

The possible values for '**result**' are:

Value	Description
OK	OK
WARNING	Informative control failed
ERROR	Blocking control failed

Category

Transaction details.

vads_risk_primary_warranty

Description

Concerns donations only (or secondary transactions).

Allows to override the risk management configuration **Verification of transfer of responsibility for primary transactions**.

Format

string (enum)

Error code

117

Possible values

Value	Description
DEFAULT or empty	Default value. Applying shop settings.
IGNORE	Ignore the value of transfer of responsibility for the primary transaction before creating a secondary transaction.
CHECK	Force the verification process of transfer of responsibility for the primary transaction before creating a secondary transaction.

Category

Donation details.

vads_sequence_number

Description

Field returned in the response.

Contains the transaction sequence number.

This field's value is always set to 1 in case of immediate payments (vads_payment_config=SINGLE).

For an installment payment, this field will take the value 1 for the first installment, 2 for the second one, 3 for the third one, etc.

Note:

vads_sequence_number *is not returned in the response when a payment is canceled or abandoned.*

Category

Transaction details.

vads_ship_to_city

Description

Allows to specify the city for shipping.

Format

an..128

Error code

83

Category

Shipping details.

vads_ship_to_country

Description Allows to define the shipping buyer's country code in compliance with the ISO 3166 standard.

Format a2

Error code 86

Examples of possible values

Country	Code
BRAZIL	BR
CORSICA	FR
FRANCE	FR
GUADELOUPE	GP
MARTINIQUE	MQ
NEW CALEDONIA	NC
SAINT PIERRE AND MIQUELON	PM
POLYNESIE FRANCAISE	PF

Category Shipping details.

vads_ship_to_delivery_company_name

Description Allows to define the name of the transporter.

Format ans..127

Error code 96

Category Shipping details.

vads_ship_to_district

Description Allows to define the shipping district.

Format ans..127

Error code 115

Category Shipping details.

vads_ship_to_first_name

Description Allows to specify the buyer's first name for shipping.

Format ans..63

Error code 106

Category Shipping details.

vads_ship_to_last_name

Description Allows to specify the buyer's last name for shipping.

Format ans..63

Error code 107

Category Shipping details.

vads_ship_to_legal_name

Description	Legal name of the shipping recipient.
Format	an..100
Error code	125
Category	Shipping details.

vads_ship_to_name

Description	Allows to specify the shipping buyer's last name.
Format	ans..63
Error code	80
Category	Shipping details.

vads_ship_to_phone_num

Description	Allows to specify the shipping buyer's phone number.
Format	ans..32
Error code	87
Category	Shipping details.

vads_ship_to_speed

Description	Allows to specify the shipping mode.
Format	string (enum)
Error code	95
Possible values	**STANDARD**, **EXPRESS**
Category	Shipping details.

vads_ship_to_status

Description	Allows to specify the type of the shipping address.
Format	string (enum)
Error code	93
Possible values	**PRIVATE**, **COMPANY**
Category	Shipping details.

vads_ship_to_state

Description	Allows to specify the buyer's state for shipping.
Format	ans..127
Error code	84
Category	Shipping details.

vads_ship_to_street

Description Allows to specify the buyer's address.

Format ans..255

Note: ***>*** *and* ***<*** *>and <are not authorized.*

Error code 81

Category Shipping details.

vads_ship_to_street2

Description Allows to specify the second line of the buyer's address.

Format ans..255

Note: ***>*** *and* ***<*** *>and <are not authorized.*

Error code 82

Category Shipping details.

vads_ship_to_street_number

Description Allows to specify the shipping street number.

Format an..5

Error code 114

Category Shipping details.

vads_ship_to_type

Description Allows to specify the shipping type.

Format string (enum)

Error code 94

Possible values

- **RECLAIM_IN_SHOP** for picking up the item at the shop.
- **RELAY_POINT** for using a third-party pick-up network (Kiala, Alveol, etc).
- **RECLAIM_IN_STATION** for picking up the item in an airport, a guard or a travel agency.
- **PACKAGE_DELIVERY_COMPANY** for shipping by the transporter (Colissimo, UPS, etc.).
- **ETICKET** for sending an electronic ticket, download.

Category Shipping details.

vads_ship_to_user_info

Description	Information about the user at the source of the payment. This parameter will be resent with the response and will include the value transmitted in the request. ***Note:*** *For backward compatibility, it is possible to use this field to set the CPF/CNPJ (legal identifier in a numeric format between 11 and 20 digits long) required by the ClearSale risk management module. However,* ***vads_cust_national_id*** *field can be used.*
Format	ans..255
Error code	116
Category	Shipping details.

vads_ship_to_zip

Description	Allows to specify the buyer's postal code.
Format	an..64
Error code	85
Category	Shipping details.

vads_shipping_amount

Description	Allows to enter the shipping fees for the whole order.
Format	n..12
Error code	109
Category	Order details.

vads_shop_name

Description	Allows to define the name of the shop that appears in the payment confirmation e-mails.
Format	ans..127
Error code	72
Category	Customization of the payment page.

vads_shop_url

Description	The shop URL that appears on the payment page and in payment confirmation e-mails. This parameter allows to override your shop's default value.
Format	ans..127
Error code	73
Category	Customization of the payment page.

vads_site_id

Description

Mandatory parameter.

Generated while subscribing to the payment gateway.

Its value can be seen in the interface of the Back Office in **Settings** > **Shop** > **Certificates** tab by all authorized persons.

If the value is not correct, when paying the buyer will get an error message on his browser.

The payment is then impossible and the transaction is definitely stopped.

A warning e-mail is then sent to the administrator contact. It contains the form that the payment gateway could not process.

Format n8

Error code 02

Category Technical details.

vads_subscription

Description

Optional parameter used for creating a subscription. It designates the ID of the subscription ID to create.

Two options are possible:

- The payment gateway manages the IDs:

  In this case, this parameter must not be populated.

  In case the subscription is successfully created, the response will contain the value generated by the payment gateway.

- The merchant website manages the IDs:

  In this case, this parameter must be populated with the desired value of the subscription ID. **The merchant website must make sure that each identifier is unique**. Any registration request containing an existing identifier will be rejected and an error message will appear.

Format ans..50

Error code 63

Category Subscription details.

vads_sub_amount

Description

Mandatory parameter used for creating a subscription.

It refers to the amount of each installment except the ones that will be defined by **vads_sub_init_amount_number**.

The value must be presented in the smallest currency unit (cents for Euro).

Example: for a transaction of 10 Euros 28 cents, the value of the parameter is 1028.

Format n..12

Error code 65

Category Subscription details.

vads_sub_currency

Description

Mandatory parameter used for creating a subscription.

Numerical code of the currency used for the subscription in compliance with the ISO 4217 standard.

Format n3

Examples of possible values

The possible currencies are:

Currency	The ISO 4217 standard	Compatible with multi-currency option	Currency	The ISO 4217 standard	Compatible with multi-currency option
Thai Baht	764	✓	Moroccan Dirham	504	
Danish Crown	208	✓	Taiwan New Dollar	901	
Icelandic Crown	352		New Turkish Lira	949	✓
Norwegian Crown	578	✓	Kuwaiti Dinar	414	
Swedish Crown	752	✓	Tunisian Dinar	788	
Czech Crown	203	✓	Argentine Peso	032	
Australian Dollar	036	✓	Mexican Peso	484	
Canadian Dollar	124	✓	Philippine Peso	608	
Hong Kong Dollar	344	✓	South-African Rand	710	✓
Singapore Dollar	702	✓	Brazilian Real	986	✓
US Dollar	840	✓	Chinese Yuan (Renminbi)	156	✓
New Zealand Dollar	554	✓	Malaysian Ringgit	458	
Euro	978	✓	Russian Ruble	643	
Hungarian Forint	348	✓	Indonesian Rupiah	360	
CFP Franc	953	✓	South Korean Won	410	
Swiss Franc	756	✓	Japanese Yen	392	✓
Indian Rupee	356	✓	Polish Zloty	985	✓
Pound Sterling	826	✓			

Error code 67

Category Subscription details.

vads_sub_desc

Description

Mandatory parameter used for creating a subscription.

It designates the subscription rule to be applied.

The expected value for this parameter is a chain of characters that respect the **iCalendar** (Internet Calendar) specification, described in RFC5545 (see http://tools.ietf.org/html/rfc5545).

Among other aspects, this specification allows to define complex subscription rules via the **RRULE** property.

For technical reasons, it is not possible to define subscription periods that are shorter than one day.

The keywords " SECONDLY" / "MINUTELY" / "HOURLY" are not taken into account.

Examples:

- To program installment payments taking place on the last day of each month for 12 months, the rule is:

  RRULE:FREQ=MONTHLY;BYMONTHDAY=28,29,30,31;BYSETPOS=-1;COUNT=12

  This rule means that if the current month does not have 31 days, the machine will take the 30th into account. If the 30th does not exist, the machine will take the 29th into account, and so on until the 28th.

  Another version of this rule: **RRULE:FREQ=MONTHLY;COUNT=5;BYMONTHDAY=-1**

- To program installment payments on the 10th of each month for 12 months, the rule is: **RRULE:FREQ=MONTHLY;COUNT=12;BYMONTHDAY=10**

- To program installment payments every three months up to December 31st, 2016:
  RRULE:FREQ=YEARLY;BYMONTHDAY=1;BYMONTH=1,4,7,10;UNTIL=20161231

  The installment payments will be due on the first day of January, April, July and October each year. The total number of installments depends on the subscription start date (see the **vads_sub_effect_date** parameter).

  For more information and examples, visit *http://recurrance.sourceforge.net/*.

Format	string
Error code	64
Category	Subscription details.

vads_sub_effect_date

Description	Mandatory parameter used for creating a subscription that allows to define the subscription start date. For a SEPA direct debit, the due date must be at least 14 calendar days from the submission date of the form (vads_trans_date). This parameter does not always match with the date of the first installment that depends only on the **vads_sub_desc** parameter. The effective date indicates from which day the subscription starts. The date format is **AAAAMMJJ**. **Example**: for February 1st 2015, enter 20150201.
Format	n8
Error code	69
Category	Subscription details.

vads_sub_init_amount

Description	Optional parameter used for creating a subscription. Amount of the first installments. The number of these first installments is specified by the **vads_sub_init_amount_number** parameter. This amount is presented in the currency defined by the **vads_sub_currency** parameter in its smallest unit (cents for Euro). Example: for a transaction of 10 Euros 28 cents, the value of the parameter is 1028.
Format	n..12
Error code	66
Category	Subscription details.

vads_sub_init_amount_number

Description	Optional parameter used for creating a subscription. Number of installments for which the **vads_sub_init_amount** amount should be applied. Once these installments will have expired, the **vads_sub_amount** amount will be used. **Example**: to define a subscription with the first 3 installments of 25 Eurosand the rest of the installments of 30 Euros, the following values will be used: • vads_sub_currency = 978 • vads_sub_init_amount_number = 3 • vads_sub_init_amount = 2500 • vads_sub_amount = 3000
Format	n..3
Code erreur	68
Category	Subscription details.

vads_tax_amount

Description Parameter that allows to define the amount of taxes for the entire order.

This amount will be presented in the smallest unit of the currency. Cents for Euro.

Concerns only the PayPal payment method.

Format n..12

Error code 108

Category Order details.

vads_theme_config

Description Allows to customize certain elements of the payment page: logos, headers and certain messages (some functions require a subscription to a **commercial option**).

This parameter contains a list of keywords (codes) associated with the elements of the payment pages (labels, images), to which values are associated.

Example:

vads_theme_config=

SUCCESS_FOOTER_MSG_RETURN=Return to the website ;CANCEL_FOOTER_MSG_RETURN=Cancel and return to the website

The basic functions are presented in the chapter **Customizing the payment page** of the **Payment form implementation guide**.

The advanced functions (subscription required) are presented in **Advanced customization of the payment page**.

Format Code1=Value1;Code2=Value2

Error code 32

Possible values

Code	Description
SUCCESS_FOOTER_MSG_RETURN	Label replacing **Return to shop** after a successful payment.
CANCEL_FOOTER_MSG_RETURN	Label replacing **Cancel and return to shop** when selecting the payment method, entering payment details and in case of a declilned payment.
SECURE_ MESSAGE	Default values: *The address of this website starting with https indicates that you are on a secure page and can safely proceed to your payment*.
SECURE_MESSAGE_REGISTER	Default value: *The address of this website starting with https indicates that you are on a secure page and can safely enter your bank details.*
SITE_ID_LABEL	Default value: *Merchant ID*
CSS_FOR_PAYMENT **CSS_FOR_PAYMENT_MOBILE** **HEADER_FOR_MAIL** **FOOTER_FOR_MAIL** **SHOP_LOGO**	See **Advanced customization of the payment page**.

Category Customization of the payment page.

vads_threeds_cavv

Description

Designates the cardholder authentication through the ACS. It is populated by the 3DS authentication server (ACS) when the buyer has been correctly authenticated (vads_threeds_status equals "Y" or "A").

Format ans..28

Error code 52

Category 3DS Authentication.

vads_threeds_cavvAlgorithm

Description

Algorithm used by the ACS to generate the CAVV value.

It is populated by the 3DS authentication server (ACS) when the buyer has been correctly authenticated (vads_threeds_status equals "Y" or "A").

Format n1

Error code 55

Possible values

Value	Description
0	HMAC
1	CVV
2	CVV_ATN
3	MasterCard SPA

Category 3DS Authentication.

vads_threeds_eci

Description

Designates the e-commerce index.

Populated by the 3DS authentication server (ACS) when the buyer has been correctly authenticated (vads_threeds_status equals "Y" or "A").

	status =Y	status = A	status = U	status =N
VISA	5	6	7	-
MasterCard	02	01	-	-

Format n..2

Error code 53

Category 3DS Authentication.

vads_threeds_enrolled

Description

Designates the cardholder's enrollment status. Populated by VISA and MASTERCARD (DS) servers during the 3D Secure authentication.

Format a1

Error code 51

Possible values

Value	Description
Y	Cardholder enrolled, 3DS authentication possible.
N	Cardholder not enrolled.

Value	Description
U	Unable to verify the cardholder's enrollment status.

Category 3DS Authentication.

vads_threeds_exit_status

Description Field returned in the response.

Final status of 3D Secure authentication.

Populated by the payment gateway.

Format n..2

Possible values

Value	Description	Value	Description
0	Initial status	13	Timeout when connecting to DS.
1	Undefined	14	This status is continued to be sent
2	Status non-applicable (integrator disabled)	16	3DS not available for this payment channel (payment by file)
3	Not an e-commerce payment	25	Merchant not enrolled, 3DS unavailable*
4	Payment without 3DS (payment by ID, PayPal, Cetelem, etc.)	26	A technical error has occurred during 3DS authentication, 3DS unavailable*
5	Merchant not enrolled, 3DS unavailable	27	Cardholder not enrolled, 3DS unavailable*
6	A technical error has occurred during 3DS authentication, 3DS unavailable	28	Invalid signature*
7	**Cardholder not enrolled, 3DS unavailable**	29	Problem caused by the ACS*
8	Invalid signature	30	The 3DS authentication has been successfully completed*
9	Problem caused by the ACS	32	Problem caused by DS*
10	**The 3DS authentication has been successfully completed**	33	Timeout while connecting to DS*
11	The 3DS authentication has been completed via the integrator	98	Initialization of 3DS authentication OK
12	Problem caused by DS	99	Unknown status

*These statuses concern 3DS payments without card details entry (payment by token).

Category 3DS Authentication.

vads_threeds_mpi

Description

Enable/Disable 3DS authentication for an e-commerce payment.

Format

n1

Error code

50

Possible values

Value	Description
absent or empty	3DS authentication managed by the payment gateway (configuration by the merchant).
0	3DS authentication managed by the payment gateway (configuration by the merchant).
1	3DS authentication completely managed by the merchant on the condition that the vads_card_number field is populated (card data entered by the merchant). In this case, the data produced by 3D Secure authentication initiated by the MPI must be submitted in specific fields of the form (vads_threeds_enrolled, vads_threeds_cavv, vads_threeds_eci, vads_threeds_xid, vads_threeds_cavvAlgorithm, vads_threeds_status).
2	3DS authentication disabled for the transaction independently of the usual configuration of the merchant.

Category

3DS Authentication.

vads_threeds_sign_valid

Description

Field returned in the response.

Designates the validity of the PARes message signature. Populated by the payment gateway.

Format

n1

Possible values

Value	Description
Empty	3DS unavailable.
0	Incorrect signature.
1	Correct signature.

Category

3DS Authentication.

vads_threeds_status

Description Defines the cardholder's authentication status. Populated by the 3DS authentication server (ACS) during the 3D Secure authentication.

Format a1

Error code 56

Possible values

Value	Description
Y	Successful authentication.
N	Authentication error.
U	Authentication impossible.
A	Authentication attempt.

Category 3DS Authentication.

vads_threeds_xid

Description Designates the unique 3DS authentication reference.

Populated by the authentication server (ACS) during the 3D Secure authentication.

Format ans..28

Error code 54

Category 3DS Authentication.

vads_trans_date

Description Mandatory parameter.

Corresponds to the time and date settings in the YYYYMMDDHHMMSS format.

This value must correspond to the current date and hour in the GMT+0 (UTC) format on a 24-hour clock.

Note: If you are using web services, the vads_trans_date value will correspond to the **transmissionDate** field. It is recommended to store this value in the database to be able to set the correct value for transmissionDate when you make calls via web services.

Format n14

Error code 04

Frequent errors:

- The date is not sent in the YYYYMMDDHHMMSS format (year, month, day, hour, minute, second).
- The date is not set in the UTC format (Coordinated Universal Time).

  Make sure you use date functions in your programming language that will generate a UTC hour (e.g.: gmdate in PHP).
- The hour must be calculated in a 24-hours format and not in a 12-hours clock.
- The buyer has waited for too long before clicking on **Pay**.
- The buyer was using browser history.

Category Transaction details.

vads_trans_id

Description	Mandatory parameter. Consists of 6 numeric characters and must be unique for each transaction for a given shop on a given day. The merchant must guarantee that it is unique within that day. It must be between 000000 and 899999. The interval between 900000 and 999999 is used for refunds and operations performed from the Back Office.
Format	n6
Error code	03 **Frequent errors**: The form will be rejected if: • The transmitted value is shorter than 6 digits. • The value is null. • The field is absent. • An identical transaction number has already been sent the same day. When the buyer clicks on "Cancel and return to shop", the transaction number will have to be different on the next attempt and the previous one will be considered as already used. If it is the same, the message "The transaction has been canceled" will appear.
Category	Transaction details.

vads_trans_status

Description

Field returned in the response.

Allows to define the transaction status.

Format

string (enum)

Possible values

Value	Description
ABANDONED	**Abandoned** The payment has been abandoned by the buyer. The transaction has not been created and **cannot be viewed in the Back Office.**
AUTHORISED	A pre-notification has been sent. The direct debit is waiting to be captured
AUTHORISED_TO_VALIDATE	The direct debit request has been made in "manual validation" mode. The merchant must manually validate the transaction in the Back Office to allow the mandate to be sent to the bank. The pre-notification will only be sent after the transaction has been validated. The transaction can be validated as long as the capture date has not passed. If the capture date has passed, the payment status changes to **EXPIRED**. The **Expired** status is final..
WAITING_AUTHORISATION	The pre-notification has not been sent yet. The direct debit request has been made with a capture delay greater to 14 days while the pre-notification delay in the contract is 14 days.
NOT_CREATED	**Transaction not created** The transaction has not been created and **cannot be viewed in the Back Office.**
INITIAL	**Pending** This status is specific to all the payment methods that require a form integration with redirection, in particular **SOFORT BANKING** and **3xCB COFINOGA**. This status is returned when: • no response is returned from the acquirer or • the acquirer response time is greater than the session payment time on the payment gateway. This status is temporary. The final status will be returned once the synchronization has been made.

Table 51: Values associated with vads_trans_status

Category

Transaction details.

vads_trans_uuid

Description

Unique transaction reference generated by the payment gateway when creating a payment transaction.

Guarantees that each transaction is unique.

Format

ans32

Category

Transaction details.

vads_url_cancel

Description

URL where the buyer will be redirected after having clicked on **Cancel and return to shop** before proceeding to payment.

Format

ans..127

Error code

27

Category

Redirection to the merchant website.

vads_url_check

Description

URL of the page to notify at the end of payment. Overrides the value entered in the notification rules.

Note

This field should be used only in exceptional cases since:

- *this URL will only be used when calling the IPN URL,*
- *the overloaded value will not be used if an automatic replay takes place.*

  It is not compatible with the execution of the request sent to the IPN from the Back Office. The called URL is the URL that has been set up in the notification rule (see chapter **Setting up notifications** of the Payment Form Implementation Guide).

Format

ans..127

Error code

33

Category

Redirection to the merchant website.

vads_url_check_src

Description

Field returned in the response.

This parameter defines the source of the notification (also called IPN).

Format

string (enum)

Possible values

Value	Description
PAY	Payment creation by form.
BO	Execution of the notification URL from the Back Office.
BATCH_AUTO	Authorization request on a payment that was waiting for authorization.
REC	Payment resulting from a subscription.
MERCH_BO	Operation processed from the Back Office.
RETRY	Automatic retry of the notification URL .

Category

Redirection to the merchant website.

vads_url_error

Description

URL where the buyer will be redirected in case of an internal processing error.

Format

ans..127

Error code

29

Category

Redirection to the merchant website.

vads_url_referral

Description

Deprecated field. Use **vads_url_refused**.

URL where the buyer will be redirected in case of a declined authorization (code 02: "Contact the card issuer") after having clicked on **Return to shop**.

Format

ans..127

Error code

26

Category

Redirection to the merchant website.

vads_url_refused

Description

URL where the buyer will be redirected in case of a declined payment after having clicked on **Return to shop**.

Format

ans..127

Error code

25

Category

Redirection to the merchant website.

vads_url_return

Description	Default URL to where the buyer will be redirected after having clicked on **Return to shop**, if **vads_url_error**, **vads_url_refused**, **vads_url_success** or **vads_url_cancel** is not set. If this field has not been transmitted, the Back Office configuration will be taken into account. It is possible to set up return URLs in TEST and PRODUCTION modes. These fields are called **Return URL of the shop in test mode** and **Return URL of the shop in production mode**; they can be viewed in **Settings** > **Shop** > **Configuration**. If no URL has been specified in the Back Office and in the form, the **Return to shop** button will redirect the buyer to the merchant website URL (**URL** field in the shop configuration section).
Format	ans..127
Error code	28
Category	Redirection to the merchant website.

vads_url_success

Description	URL where the buyer will be redirected in case of a successful payment after having clicked on **Return to shop**.
Format	ans..127
Error code	24
Category	Redirection to the merchant website.

vads_user_info

Description	Information about the user at the source of the payment. For payments by form, this parameter will be resent in the response and will include the value transmitted in the request. For manual payments from the Back Office, this field will be populated with the login of the user who processed the payment. ***Note:*** *For backward compatibility, it is possible to use this field to set the CPF/CNPJ (legal identifier in a numeric format between 11 and 20 digits long) required by the ClearSale risk management module. However,* ***vads_cust_national_id*** *field can be used.*
Format	ans..255
Error code	61
Category	Buyer details

■ vads_validation_mode

Description Specifies the validation mode of the transaction.

Format string (enum)

Error code 05

Possible values

Value	Description
Absent or empty	Takes the value specified in the Back Office.
0	Automatic validation by the payment gateway.
1	Manual validation by the merchant.

Category Transaction details.

■ vads_version

Description Mandatory parameter.

Version of the exchange protocol with the payment gateway.

Format string (enum)

Error code 01

Possible value V2

Category Technical details.

■ vads_warranty_result

Description Field returned in the response.

Indicates the payment guarantee in case the payment has been accepted.

Format string (enum)

Possible values

Value	Description
YES	The payment is guaranteed.
NO	The payment is not guaranteed.
UNKNOW	Due to a technical error, the payment cannot be guaranteed.
Not populated.	Payment guarantee not applicable.

Category Transaction details.