



Managing Payments by Token
Implementation guide - File exchange

Document version 2.2

Contents

1. HISTORY OF THE DOCUMENT.....	3
2. GETTING IN TOUCH WITH TECHNICAL SUPPORT.....	4
3. GENERAL PRINCIPLE.....	5
3.1. Service presentation.....	5
3.2. Operating principle.....	5
4. MANAGING PAYMENT FILES.....	7
4.1. Generating a payment file (REQ).....	7
4.2. Uploading payment files.....	9
4.3. Downloading the return file.....	10
4.4. Analyzing the return file (ANS).....	10
5. NOTIFICATIONS.....	13
5.1. Instant Payment Notification URL at the end of payment.....	13
5.2. Confirmation e-mail of payment.....	13
6. TEST OF PAYMENT BY TOKEN.....	14
6.1. Signing in to the Merchant Back Office.....	14
6.2. Understanding the working principle.....	14
Checking the file format.....	15
Importing a payment file.....	17
6.3. The test payment result.....	18
7. DATA DICTIONARY.....	20

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
2.2	Natixis Solutions Payment	18/04/2019	<ul style="list-style-type: none"> • Addition of information on the maximum recommended size of input files (REQ). • Addition of information in the operating principle of _ERROR and _DUPLICATE files. • Time zone specified across the entire document. • Correction of line 14 of the detailed record format in the return file (ANS).
2.1	Natixis Solutions Payment	07/01/2019	Additional details on return codes in the chapter Analyzing the return file
2.0	Natixis Solutions Payment	02/01/2017	Addition of chapters on testing the payment by identifier (token).
1.9	Natixis Solutions Payment	30/03/2016	Corrected position of <i>Order details 3</i> in the chapter Generating a payment file
1.8	Natixis Solutions Payment	17/12/2015	Addition of the chapter Data dictionary
1.7	Natixis Solutions Payment	19/06/2015	Update of the chapter Generating a payment file
1.6	Natixis Solutions Payment	28/05/2015	Initial version in DITA format
1.5	Natixis Solutions Payment	20/06/2013	<ul style="list-style-type: none"> • Addition of authorization return codes • Additional details on providing the ANS response file • Update of notifications (removal of the call to the IPN URL)
1.4a	Natixis Solutions Payment	11/12/2012	Update of field 25
1.4	Natixis Solutions Payment	31/05/2012	<ul style="list-style-type: none"> • Update of the Payment file chapter • V2: Addition of order ref. / Info1 / Info2 / Info3 in the return file
1.2	Natixis Solutions Payment	07/01/2011	Additional details on the server to server response
1.1	Natixis Solutions Payment	03/12/2010	Addition of content examples of REQ and ANS files
1.0	Natixis Solutions Payment	10/05/2010	Initial version.

This document and its contents are confidential. It is not legally binding. No part of this document may be reproduced and/or forwarded in whole or in part to a third party without the prior written consent of Natixis Payment Solutions. All rights reserved.

2. GETTING IN TOUCH WITH TECHNICAL SUPPORT

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

by phone at:

0810004724

Service fee 0,06 € / min
+ call charge

for the clients of Banque Populaire

by phone at:

0811363364

Service fee 0,06 € / min
+ call charge

for the clients of Caisse d'Epargne

by e-mail:

supportvad@lyra-network.com

via your Merchant Back Office:

(Menu: **Help** > **Contact support**)

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

This information is available in the "registration of your shop" e-mail or in the Merchant Back Office (**Settings** > **Shop** > **Configuration**).

3. GENERAL PRINCIPLE

3.1. Service presentation

The Exchange File service of the Management of payment by identifier feature allows merchant websites to execute debit operations on the bank cards of their registered customers.

The operations will be executed by the service in the form of "batch treatment" : the merchant website sends to the payment gateway a series of orders in the form of files. The payment gateway treats these orders and in turn generates reporting files.

3.2. Operating principle

The used file transfer protocol is SFTP.

In SFTP transactions, the payment gateway acts as the server, and the merchant website as the client.

The access to the directory where the files will be stored is protected by the username/password combination that was sent to you when the service was first used.

Persistent data

The files uploaded by the merchant to the FTP server or generated by the payment gateway are compressed after 7 days.

The **.gz** extension is added to the filename.

The files are definitively deleted as soon as their creation date exceeds 15 days.

This removal will always be initiated by the payment gateway.

The service is broken down into three phases:

Phase of scanning of the uploaded files

An automatic task allows to retrieve all the files uploaded to the FTP server in order to transmit them to the payment gateway for validation.

Validation phase

Several checkups are made before processing the files.

- The naming rules specific to each file type must be applied. If there is a file that does not respect these naming rules, the validation process ends and the **_ERROR** suffix is added to the filename.
- The file size is analyzed.

If there is a file whose size is 0 bytes, the validation process ends and the **_ERROR** suffix is added to the filename.

- The payment gateway records the processed valid filenames.

If there is a file that has already been processed, the validation process ends and the **_DUPLICATE** suffix is added to the filename.

- The files with a name that contains a **_DUPLICATE**, **_ERROR** suffix or the **“.gz”** extension are ignored.

Invalid files are stored in the **depot** folder until they are deleted by the merchant or automatically.

Processing phase

When a valid file is processed, the input file (REQ) is moved to the **id** directory.

A response file (ANS) is created in the **id** directory.

If an anomaly is detected while the files are processed, a warning e-mail is sent to the merchant.

Note

All the times indicated in this document are based on Paris local time.

4. MANAGING PAYMENT FILES

Several steps are required for using the file exchange service.

The steps are:

- Generate a payment request file
- Upload a payment request file
- Obtain a response file
- Analyze a response file

These steps are described in the following chapters.

4.1. Generating a payment file (REQ)

The payment request file consists of a series of records.

A “record” is a list of parameters separated by the “;” character ending with a carriage return. Thus, there is one record on every line.

This format type is generally called "CSV" (Comma Separated Values), and is 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.



The number of records present in the files directly affects the amount of time it will take to process the file as well as the availability of the return file.

Beyond 2,000 records it is recommended to generate several payment files.

Example:

A test file was submitted on 30/11/2018 for the 12345678 store. It contains two payment requests with the following parameters:

First payment by token:

- ID: 'TEST-TOKEN-1'
- Transaction number: 000001
- Amount: 935,99 EUR

Second payment by token:

- ID: 'TEST-TOKEN-2'
- Transaction number: 000002
- Amount: 75,90 EUR

1. The name of the file must comply with 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 (2 numeric characters) of the file within the same day.

The name of the file in the example is 20181130.12345678.PAY.REQ.T.01

2. Populate the header record with the following values:

Pos.	Description	Format	Values	Equivalent
01	Record code	n2	00	
02	File type		PAY	
03	File version	n2	02	
04	Shop ID	n8	12345678	vads_site_id
05	Environment (TEST or PRODUCTION)		TEST	vads_ctx_mode
06	Creation date	n8	In YYYYMMDD format	
07	Time of creation	n6	In 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 YYYYMMDD format	vads_trans_date
04	Transaction time	n6	In HHMMSS format	vads_trans_date
05	Transaction identifier	n..6	Unique within the same day and shared with other payment modes	vads_trans_id
06	Transaction type	a2	CD	
07	Amount	n..12	In its smallest currency unit.	vads_amount
08	Currency code	n3	E.g.: 978 for euro (EUR)	vads_currency
09	Requested capture date	n8	In YYYYMMDD format 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	Card token	ans..50	Card token to use for the direct debit	vads_identifier
12	Merchant ID to be used.	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
16	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
01	Record code	n2	01
02	Number of transmitted detail records	n..6	E.g.: 2. It is recommended to not exceed 2,000 records per file.

Example of a file:

File name: 20181130.12345678.PAY.REQ.T.01

```
00;PAY;02;12345678;TEST;20181130;102008;  
02;1;20181130;102008;000001;CD;93599;978;;;TEST-TOKEN-1;;;;  
02;2;20181130;102008;000002;CD;7590;978;;;TEST-TOKEN-1;;;;  
01;2
```

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

It allows:

- to check the structure of the file to be submitted,
- to import the TEST payment file,
- to create TEST payments on the basis of the imported file.

4.2. Uploading payment files

The payment platform processes the payment files transferred by the merchant website twice **every day**: once at **7 a.m.** and once at **1 p.m.**, Europe/Paris time zone.

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

1. Connect to the FTP server:

In SFTP:

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

2. Upload your file into the folder **[Shop ID] > depot.**

4.3. Downloading the return file

The payment platform processes the payment files transferred by the merchant website twice **every day**: once at **7 a.m.** and once at **1 p.m.**, Europe/Paris time zone.

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.**, Europe/Paris time zone.

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 (2 numeric characters) of the file within the same day.

Check the presence of an error file (**_ERROR** or **_DUPLICATE**) in the **depot** directory if you cannot find your return file in the **id** directory.

4.4. Analyzing the return file (ANS)

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 payment rejection, the source of the error will be indicated in the detail 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 re-submit the file 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 re-submit the unprocessed payments in a new file.

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

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

Its content is:

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

Format of the header:

Pos.	Description	Format	Values	Equivalent in the data dictionary
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.	
06	Shop ID	n8	Same as in the request, if valid.	vads_site_id
07	Environment		Same as in the request, if valid.	vads_ctx_mode
08	Creation date	n8	Same as in the request, if valid.	
09	Time of creation	n6	Same as in the request, if valid.	
10	End date of processing In YYYYMMDD format	n8	20161130	
11	End time of processing	n6	130019	

Format of detailed record:

Pos.	Description	Format	Values	Equivalent in the data dictionary
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	CD	
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	vads_currency
09	Countervalue of the amount in euro	n..12	Same as the value used in the request, if valid	vads_effective_amount
10	Countervalue currency	n3	Same as the value used in the request, if valid	

Pos.	Description	Format	Values	Equivalent in the data dictionary
11	Requested capture 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	Token	ans..50	Same as the value used in the request, if valid	vads_identifier
14	Merchant ID	n7	MID 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		Examples: <ul style="list-style-type: none"> • 00: Successfully processed – Payment accepted • 05: Successfully processed – Payment rejected • 30: Parameter error. Additional information in provided in field 25 • 96: Technical error 	vads_result
20	Authorization return code	an..2	Value returned by the authorization server. See the list of codes in the data dictionary chapter. Empty if not applicable.	vads_auth_result
21	Authorization number returned by the acquirer	n6	not applicable	vads_auth_number
22	Authorization mode		FULL	vads_auth_mode
23	Authorization date	n8	In YYYYMMDD format	
24	Authorization hour	n6	In 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. See the list of codes in the data dictionary chapter. Empty if not applicable.	vads_extra_result
26	Payment method number (Card number or IBAN and BIC number)	an..36	Extract of the payment method number. (E.g.: IBAN_BIC) The BIC is optional so the number may just be the IBAN.	vads_card_number
27	Expiry date	n8	In YYYYMMDD format Empty if not applicable	

Format of the trailer record:

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

5. NOTIFICATIONS

5.1. Instant Payment Notification URL at the end of payment

The IPN won't be call during file processing.

Only the analysis of the response file (made available in the /id directory) should allow the merchant to update his information system.

5.2. Confirmation e-mail of payment

No confirmation e-mail won't be sent by the payment gateway to the buyer or to the merchant in case of accepted payment.

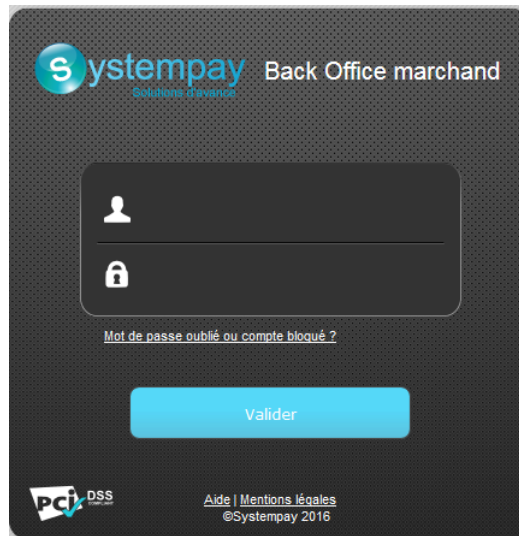
No confirmation e-mail won't be sent by the payment gateway to the merchant in case of declined payment.

6. TEST OF PAYMENT BY TOKEN

6.1. Signing in to the Merchant Back Office

Sign in the Back Office:

<https://paiement.systempay.fr/vads-merchant/>



1. Enter your login.

The login is sent to the merchant's e-mail address (the subject of the e-mail is **Connection identifiers- [your shop name]**).

2. Enter your password.

The password is sent to the merchant's e-mail address (the subject of the e-mail is **Connection identifiers- [your shop name]**).

3. Click on **Validate.**

After 3 password entry errors, the user's account is locked. Click on the link **Forgotten password or locked account** to reset.

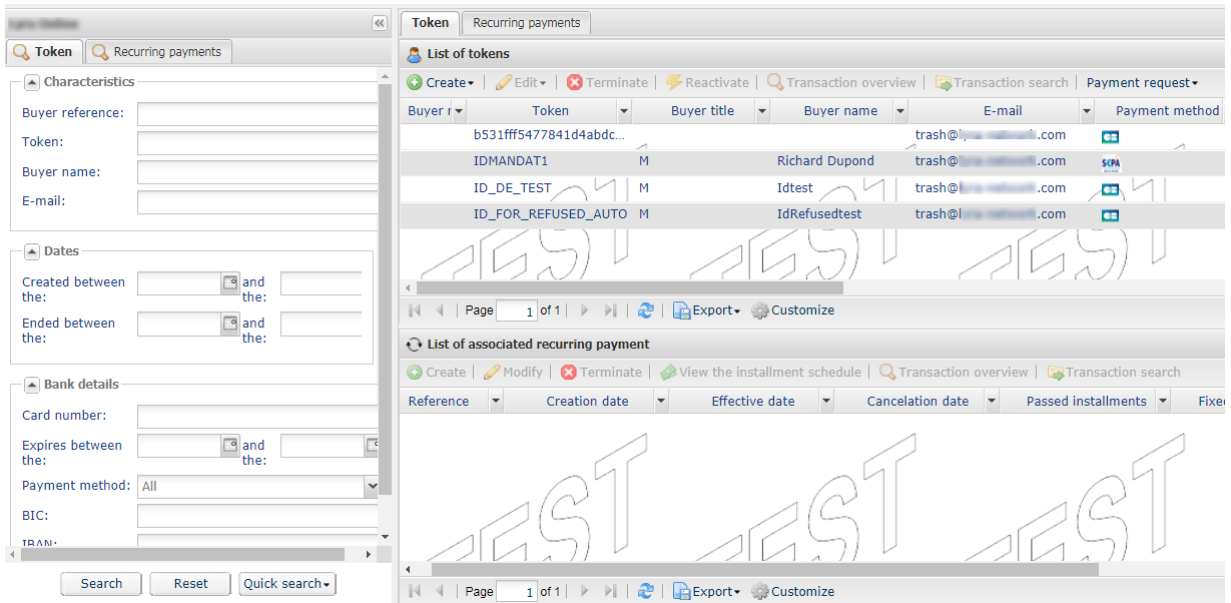
6.2. Understanding the working principle

The files uploaded to the FTP server are processed twice a day.

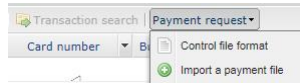
In order not to slow down the implementation of the solution for payment by token in file mode, the merchant can perform **on-demand** tests of their payment files **in TEST mode** in the Merchant Back Office.

1. Click on **Management > TEST recurring payments**

The TEST recurring payments page appears.



2. Click on **Payment request**



Two actions are available:

- Check file format.
- Import a payment file.

Checking the file format

This option allows to perform several checks on the structure of your file.

When you select **Check the file format**, the import dialog box appears.

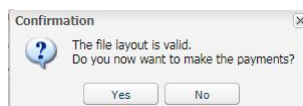
1. Click **Browse**.
2. Select the file to import.

The size of the file must not exceed 1kb.

*The file format is CSV and its name must respect the rule defined in the chapter **Generating a payment file**. Example: 20161229.91335531.PAY.REQ.T.01*

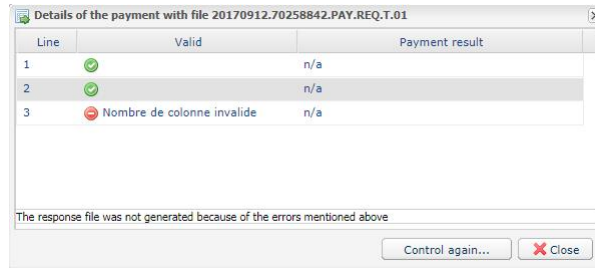
3. Click **Check**.

If the file is properly structured, the following message appears.



If the file is structured incorrectly, a report appears indicating the detected error and the line that contains it.

Example:



The tables below present the different types of errors related to structure.

- **File format error**

Error message	Description
The specified file size exceeds the maximum authorized size	The size of the file must not exceed 1kb. It is recommended to create the file using programs such as Notepad , Notepad ++ , etc.
Name of the erroneous file	Your file does not respect the naming convention as described in the chapter Generating a payment file

- Errors in the **Header record line**

Error message	Description
Expected header type (record code 00)	The first line of the file corresponds to the header record and must start with 00
Badly placed header (should be on the 1st line)	
Invalid number of columns	The header record must include 8 columns.
PAY not present in the header (null)	The field n°2 of the header record must be populated with PAY
Incorrect header version (null)	The field n°3 must be populated with 01
Incorrect shop identification (null)	The Shop ID must contain 8 digits
Incorrect environment (null)	Two possible values: TEST or PRODUCTION
Only transactions in TEST mode are possible	You cannot submit a payment file in PRODUCTION mode.
Incorrect creation date or time	The date must have the yyyyMMdd format. The time must have the HHmmSS format.
Execution date not empty	This field must not be populated. It must remain empty.
Header does not match filename	The header and the filename must contain the same values of: <ul style="list-style-type: none"> • the shop ID, • the environment, • the creation date. <p><u>Example:</u> Filename: 20161229.91335531.PAY.REQ.T.01 Header:</p> <pre>00;PAY;02;91335531;TEST;20161229;140800;</pre>

- Errors in the **Detail record line**

Error message	Description
Bad sequence number (field 2)	Must start with 1 . Given that the format (n..6) of the sequence contains 6 digits, it is safer to number the sequences 000001 , 000002 , etc. to avoid errors.
Invalid number of columns	Each detail record must include 16 columns.

- Errors in the **Trailer record line**

Error message	Description
No trailer (record code 01)	The last line of the file must contain the trailer and start with 01 .

Error message	Description
Invalid number of columns	The trailer record must include 2 columns.
Bad record number	Must be equal to the exact number of transmitted detail records. When the detail record contains: <ul style="list-style-type: none"> • One record line, the value is 1. • Two record lines, the value is 2. • Three record lines, the value is 3. • etc.

- Other technical errors

Error message	Description
Unknown line type	None of the lines appearing before the trailer record of the file must be empty. All lines must start with 00 , 01 or 02 .
Data after end of file	No lines (empty or not) must be present after the trailer record.

Importing a payment file

This option allows to perform several checks on the structure and contents of your file and make payments based on the information that it contains.

Once the file has been imported and processed, the gateway generates a response file that is sent by e-mail to the address indicated upon import.

Warning: If the file contains structure errors, it will be rejected and no reports will be generated. Therefore, it is important to check the file first.

When you select **Import a payment file**, the import dialog box appears.

1. Click **Browse**.
2. Select the file to import.

The size of the file must not exceed 1kb.

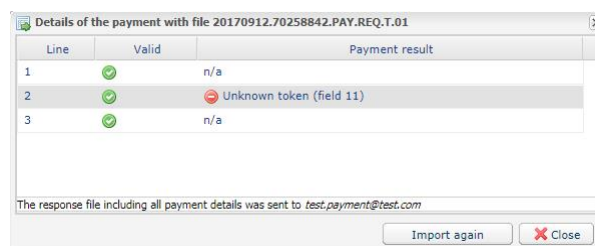
*The file format is CSV and its name must respect the rule defined in the chapter **Generating a payment file**. Example: 20161229.91335531.PAY.REQ.T.01*

3. Enter the e-mail address that will receive the payment result.
4. Click **Import**.

If the structure and the contents of the file are correct, the payment details appear without any detected errors.

If the structure and/or the contents of the file are incorrect, a report appears indicating the detected error and the line that contains it.

Example:



The tables below present the different types of errors related to structure and/or contents.

- Errors in the **Header record line**

Error message	Description
Unknown shop	The shop ID specified in the file must refer to one of the shops to which you have access with your Back Office identifiers.

- Errors in the **Detail record** line

Error message	Description
DATE_TRANSACTION format error	The date format must be yyyyMMdd.
TIME_TRANSACTION format error	The time format must be HHmmSS.
ID_TRANSACTION format error	The format of this field must contain 6 numeric characters.
The provided transaction ID is already in use	The transaction ID must be unique for each day and each shop.
TYPE_TRANSACTION format error	Must be set to CD.
AMOUNT format error (field 7)	The amount must be specified in the smallest currency unit. E.g.: 3000 for 30,00 EUR
CURRENCY format error (field 8)	3-digit numeric code in compliance with the ISO 4217 standard. E.g.: 978 for euro (EUR)
CAPTURE_DATE format error (field 9)	The date format must be yyyyMMdd.
VALIDATION_MODE format error	Possible values: <ul style="list-style-type: none"> • 0 (automatic validation), • 1 (manual validation), • empty (default configuration as defined in the Merchant Back Office)
The provided token is unknown	The tokens specified in the file must exist in TEST recurring payment mode.
TOKEN format error	The token must not be empty.
The provided Merchant ID is unknown (field 12)	This optional field must be populated.

- Errors in the **Trailer record** line

Error message	Description
No trailer (record code 01)	The last line of the file must contain the trailer and start with 01 .
Invalid number of columns	The trailer record must include 2 columns.
Bad record number	Must be equal to the exact number of transmitted detail records. When the detail record contains: <ul style="list-style-type: none"> • One record line, the value is 1. • Two record lines, the value is 2. • Three record lines, the value is 3. • etc.

- Other technical errors

Error message	Description
Unknown line type	None of the lines appearing before the trailer record of the file must be empty. All lines must start with 00 , 01 or 02 .
Data after end of file	No lines (empty or not) must be present after the trailer record.

6.3. The test payment result

Once the file has been imported and processed, the gateway generates a response file that is sent in attachment by e-mail to the address indicated during the import.

The received file is of CSV type and its size is maximum 1kb. It is named according to the rule defined in the chapter **Downloading the return file**. Example: 20161229.91335531.PAY.ANS.T.01

The body of the e-mail informs you about the transaction result following your payment request by token file.

The payment result is displayed in the **Transactions in progress** tab among the test transactions in your Merchant Back Office. You can view the transaction details of the duplicate transaction by clicking on it.

7. DATA DICTIONARY

The chapter presents the list of fields that can be used in payment by file exchange.

To view all the existing fields, please refer to the *Implementation guide Hosted Payment Page* available in our online document archive.

■ vads_amount

Description	Transaction amount expressed in the smallest currency unit (cents for euro). <u>Example</u> : for a transaction of 10 EUR and 28 cents, the value of the parameter is 1028. The payment form will be rejected in the following cases: <ul style="list-style-type: none">• 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). <i>Input and output field, returned in the response (IPN and Return URL).</i>
Format	n..12
Error code	09
Category	Transaction details.

■ vads_auth_number

Description	Authorization number returned by the authorization server, if available (otherwise, empty). <i>Output field, returned in the response (IPN and Return URL).</i>
Format	an..6
Category	Transaction details

■ vads_auth_mode

Description	Specifies the mode of the authorization request. <i>Output field, returned in the response (IPN and Return URL).</i>
Format	enum
Possible values	FULL : corresponds to an authorization for the total amount of the transaction. Value used in case of immediate payments, if the period between the requested capture date and the current date is strictly shorter than the authorization validity period. MARK : corresponds to an authorization for 1 EUR (or information request about the CB network if the acquirer supports it).

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_result

Description Return code of the authorization request returned by the issuing bank, if available.

Output field, returned in the response (IPN and Return URL).

Format an..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 for this cardholder	YES
13	Invalid amount	YES	58	Transaction not allowed for this cardholder	
14	Invalid cardholder number	YES	59	Suspected fraud	YES
15	Unknown issuer	YES	60	The acceptor of the card 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	Suspected fraud	YES			

Table 1: Return codes specific to the CB network

Code	Description
0	Approved or completed successfully
2	Call Voice-authorization number; Initialization Data
3	Invalid merchant number
4	Retain card
5	Authorization declined
10	Partial approval
12	Invalid transaction
13	Invalid amount
14	invalid card
21	No action taken
30	Format Error
33	Card expired
34	Suspicion of Manipulation
40	Requested function not supported
43	Stolen Card, pick up
55	Incorrect personal identification number
56	Card not in authorizer's database
58	Terminal ID unknown
62	Restricted Card
78	Stop payment order
79	Revocation of authorization order
80	Amount no longer available
81	Message-flow error
91	Card issuer temporarily not reachable
92	The card type is not processed by the authorization center
96	Processing temporarily not possible
97	Security breach - MAC check indicates error condition
98	Date and time not plausible
99	Error in PAC encryption detected

Table 2: Codes returned by the GICC network

Code	Description
000	Approved
001	Approved with an ID
002	Partial approval (Prepaid Cards only)
100	Rejected
101	Expired card / Invalid expiry date
106	Exceeded PIN entry 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	Cardholder not enrolled / not allowed
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

Code	Description
900	Accepted - ATC Synchronization
909	System malfunction (cryptographic error)
912	Issuer not available

Table 3: Codes returned by AMEX Global acquirer

Other return codes For payment methods that are different from the ones presented below:

- see the technical documentation specific to the payment method
or
- contact the technical support for more information.

Category Transaction 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 Merchant Back Office will be used. The default value can be configured in the Merchant Back Office by all authorized persons.

Note:

- The value of **vads_capture_delay** is not taken into account in the case of payment in installments **MULTI_EXT**.
- If the capture delay is higher than 365 days in the payment request, it will be automatically reset to 365 days.

Input field.

Format n..3

Error code 06

Category Transaction details.

■ vads_card_number

Description

- Masked card number. 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.
The BIC is optional so the number may just be the IBAN.

Output field , returned in the response (IPN and Return URL).

Format an..36

Category Payment method details.

■ vads_currency

Description Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard.

To use a currency during a payment, you must have a MID created in this currency. The acquirer provides the MID to the merchant with the supported currency(ies) and the gateway takes this information into account when creating a MID.

Input and output field, returned in the response (IPN and Return URL).

Format n3

Error code 10

Possible values

Currency	ISO 4217 encoding	Number of digits after the decimal point
Australian Dollar (AUD)	036	2
Cambodian Riel (KHR)	116	0
Canadian Dollar (CAD)	124	2
Chinese Yuan Renminbi (CNY)	156	1
Croatian Kuna (HRK)	191	2
Czech Crown (CZK)	203	2
Danish Crown (DKK)	208	2
Hong Kong Dollar (HKD)	344	2
Hungarian Forint (HUF)	348	2
Indian Rupee (INR)	356	2
Indonesian Rupiah (IDR)	360	2
Japanese Yen (JPY)	392	0
South Korean Won (KRW)	410	0
Malaysian Ringgit (MYR)	458	2
Mexican Peso (MXN)	484	2
New Zealand Dollar (NZD)	554	2
Norwegian Crown (NOK)	578	2
Philippine Peso (PHP)	608	2
Russian Ruble (RUB)	643	2
Singapore Dollar (SGD)	702	2
South-African Rand (ZAR)	710	2
Swedish Crown (SEK)	752	2
Swiss Franc (CHF)	756	2
Thai Baht (THB)	764	2
Pound Sterling (GBP)	826	2
US Dollar (USD)	840	2
Taiwan New Dollar (TWD)	901	2
Romanian Leu (RON)	946	2
Turkish Lira (TRY)	949	2
CFP Franc (XPF)	953	0
Bulgarian Lev (BGN)	975	2
Euro (EUR)	978	2
Polish Zloty (PLN)	985	2
Brazilian Real (BRL)	986	2

Currencies supported by Concardis acquirer:

Acquirer	Currencies supported
Concardis (GICC)	AUD, CAD, CZK, DKK, HUF, JPY, NOK, ZAR, SEK, CHF, GBP, USD, PLN, AED

Category Transaction details.

■ vads_ctx_mode

Description	<p>Mandatory parameter.</p> <p>Defines the mode of interaction with the payment gateway.</p> <p>Affects the choice of the key to be used (test or production) during signature computation.</p> <p>The TEST mode is available at all times, even after the generation of the production key.</p> <p>If you create a new e-commerce website (or have access to the acceptance testing environment), you can make tests without impacting the website that is currently in production.</p> <p><i>The input and output field, returned in the response (IPN and Return URL).</i></p>
Format	enum
Error code	11
	<p>Frequent errors:</p> <ul style="list-style-type: none">• The mode has not been submitted to the payment platform.• Do not code PROD instead of PRODUCTION• Do not code the value in lowercase letters (test or production). This field expects values only in uppercase letters without abbreviations.
Possible values	TEST, PRODUCTION
Category	Technical details

■ vads_effective_amount

Description Payment amount in the currency used for the capture.
Output field, returned in the response (IPN and Return URL).

Format n..12

Examples **EXAMPLE FOR A SHOP WITH CAPTURE IN EUR**

Payment of 10,00 EUR

Parameters sent in the payment form
<ul style="list-style-type: none"> vads_amount = 1000 vads_currency = 978

Returned parameters
<ul style="list-style-type: none"> vads_amount = 1000 vads_currency = 978 vads_effective_amount = 1000

A payment of 10-US Dollar

Parameters sent in the payment form
<ul style="list-style-type: none"> vads_amount = 1000 vads_currency = 840

Returned parameters
<ul style="list-style-type: none"> 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 90,00 EUR in 3 installments

Parameters sent in the payment form
<ul style="list-style-type: none"> vads_amount = 9000 vads_currency = 978 vads_payment_config=MULTI_EXT:date1=3000;date2=2000;date3=4000

Returned parameters for the first installment
<ul style="list-style-type: none"> vads_amount = 9000 vads_currency = 978 vads_effective_amount = 3000

A payment of 90-US Dollar paid in 3 installments

Parameters sent in the payment form
<ul style="list-style-type: none"> vads_amount = 9000 vads_currency = 840 vads_payment_config=MULTI_EXT:20121025=3000;20121026=2000;20121027=4000

Returned parameters for the first installment
<ul style="list-style-type: none"> 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_expiry_month

Description Expiry month of the card used for the payment.
Output field , returned in the response (IPN and Return URL).

Format n..2

Category Payment method details.

■ vads_expiry_year

Description Expiry year of the card used for the payment.
Output field , returned in the response (IPN and Return URL).

Format n4

Category Payment method details.

■ vads_extra_result

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

- If **vads_result** equals 30 (request error), then **vads_extra_result** contains the numeric 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 numeric code of the risk management result.

Code	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.

Code	Description
04	The country of origin of the card is on the merchant's greylist.
05	The IP address is on the merchant's greylist.
06	The BIN code is on 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 this IP address belongs to the greylist.
99	Technical issue encountered by the server during a local verification process.

Output field, returned in the response (IPN and Return URL).

Category Technical details.

■ vads_identifier

Description 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.

- Otherwise, it can be generated by the merchant website.

In this case, this parameter must be populated with the desired value of the identifier. **Warning, 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.

Input and output field, returned in the response (IPN and Return URL).

Format Deux formats possibles :

- **an32** : lorsque l'identifiant est généré par la plateforme
- **ans..50** : lorsque l'identifiant est généré par le marchand

Error code 30

Category Recurring payment 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.

Input and output field, returned in the response (IPN and Return URL).

Format ans..64

Error code 13

Category Order details.

■ vads_order_info

Description	Order description. <i>Input and output field, returned in the response (IPN and Return URL).</i>
Format	an..255
Error code	14
Category	Order details.

■ vads_payment_config

Description	<p>Defines the type of payment: immediate or installment.</p> <ul style="list-style-type: none"> 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 . <p>The parameters are:</p> <ul style="list-style-type: none"> "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. <p>The field order associated with MULTI must be respected.</p> <ul style="list-style-type: none"> 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. <p>The dates must not be passed</p> <p>The MULTI_EXT value requires a recurring payment to the Advanced installment payment.</p> <p><i>Note: The value of vads_capture_delay is not taken into account in the case of payment in installments MULTI_EXT.</i></p> <p>Input and output field, returned in the response (IPN and Return URL).</p>
Format	enum
Error code	07
Possible values	<ul style="list-style-type: none"> SINGLE MULTI:first= initial_amount;count=installments_nb ;period=interval_in_days MULTI_EXT:date1=amount1;date2=amount2;date3=amount3
Example 1	<p>MULTI allows to define an installment payment.</p> <p>The amount of each installment corresponds to the total amount divided by the number of installments.</p> <p>The amount of the first installment can be different, it can be specified in first parameter.</p>

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 ,00 EUR will be captured by the bank in 2 days (vads_capture_delay).

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

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

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

The total amount is 200,00 EUR (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.

MULTI_EXT : payment request:

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

Result:

The first payment of 50 ,00 EUR is scheduled for June 1st 2015..

The second payment of 25 ,00 EUR is scheduled for July 1st 2015..

The last payment of 25 ,00 EUR 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_result**

Description	Return code of the requested action. <i>Output field, returned in the response (IPN and Return URL).</i>
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 associate with the value of the vads_extra_result field.
96	Technical error.

Category Technical details.

■ **vads_site_id**

Description

Mandatory parameter.

Value generated during the subscription to the payment gateway.

Its value can be seen in the interface of the Merchant Back Office via **Settings > Shop > Keys** by all authorized individuals.

If the value is incorrect, the buyer will get an error message in their browser when making the payment.

The payment is then not possible and the transaction is permanently interrupted.

A warning e-mail is then sent to the shop administrator. It contains the form that the gateway was unable to process with the value of the signature.

Input and output field, returned in the response (IPN and Return URL).

Format n8

Error code 02

Category Technical details.

■ **vads_trans_date**

Description

Mandatory parameter.

Corresponds to the timestamp in the YYYYMMDDHHMMSS format

The timestamp must necessarily correspond to the current date and time, in the GMT + 0 (or UTC) time zone in 24h format.

Note: If you are using Web Services, the vads_trans_date field 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.

Input and output field, returned in the response (IPN and Return URL).

Format n14

Error code 04

Common mistakes:

- The date has not been submitted in the YYYYMMDDHHMMSS format (year, month, day, hour, minute, second).
- The date is not based on the UTC time zone (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 time must be calculated using the 24h format, not 12h.
- 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.

It consists of 6 numeric characters and must be unique for each transaction for a given shop on the same day.

Note: *the uniqueness of the transaction identifier is based on the universal time (UTC).*

The merchant website must guarantee this uniqueness during same the day. It must be between 000000 and 899999.

The numbers between 900000 and 999999 are reserved for refunds and operations made via the Merchant Back Office.

Input and output field, returned in the response (IPN and Return URL).

Format n6

Error code 03

Common errors:

The form is rejected:

- if the transmitted value contains less than 6 digits
- if the value is null
- if the field is absent
- if an identical transaction number has already been sent on the same day.

If the buyer clicks on "Cancel and return to the shop", the transaction number must be different for the next attempt as the previous one will be considered as already used.

Otherwise, the message "The transaction has been canceled" will appear.

Category Transaction details.

■ vads_validation_mode

Description Specifies the validation mode of the transaction

Input and output field, returned in the response (IPN and Return URL).

Format enum

Error code 05

Possible values

Value	Description
Absent or empty	Takes the value specified in the Merchant Back Office

Value	Description
0	Automatic validation by the payment gateway.
1	Manual validation by the merchant.

Category Transaction details.