COLLECTING SOLUTION

Management of payments by token

Implementation Guide - File exchange

Document version 2.3

Contents

1. HISTORY OF THE DOCUMENT....3

2. OBTAINING HELP....4

3. GENERAL PRINCIPLE....5

3.1. Presentation of the service....5

3.2. Operating principle....5

4. FORMAT OF EXCHANGED FILES....7

4.1. Format of request file....7

Naming rules....9

4.2. Format of response file....9

Naming rules....13

5. VALIDATION PROCESS OF A TEST FILE....14

5.1. Checking the file format....15

5.2. Importing a payment file....17

5.3. Analyzing the result....19

6. UPLOADING A FILE TO THE SFTP SERVER....20

7. PROVIDING THE RESPONSE FILE....21

8. ERROR HANDLING....22

9. NOTIFICATIONS....23

9.1. Instant Payment Notification URL at the end of payment....23

9.2. Confirmation e-mail of payment....23

10. DATA DICTIONARY....24

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
2.3	Lyra Collect	7/31/2020	• Document overhaul. • Addition of a warning concerning the absence of extension in the name of the request and response files. • Addition of a diagram illustrating the operating principle. • Addition of the *Error handling* chapter. • Update of network authorization return codes for the CB network.
2.2	Lyra Collect	4/18/2019	• 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	Lyra Collect	1/7/2019	Additional details on return codes in the chapter **Analyzing the return file**
2.0	Lyra Collect	10/1/2018	Initial version

2. OBTAINING HELP

Looking for help? Check our FAQ on our website

https://lyra.com/doc/fr/collect/faq/sitemap.html

If you have any technical questions or need assistance, our tech support is available

from Monday to Friday from 9 a.m. to 6 p.m.

by phone at: **0811900475** Service fee 0.06 € / min + call charge

by e-mail : *support-ecommerce@lyra-collect.com*

and via your Expert Back Office, **Help** > **Contact support**

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

3. GENERAL PRINCIPLE

3.1. Presentation of the service

The file exchange service of the payment by token management module enables merchant sites to carry out debit transactions with the bank cards of their subscribing customers.

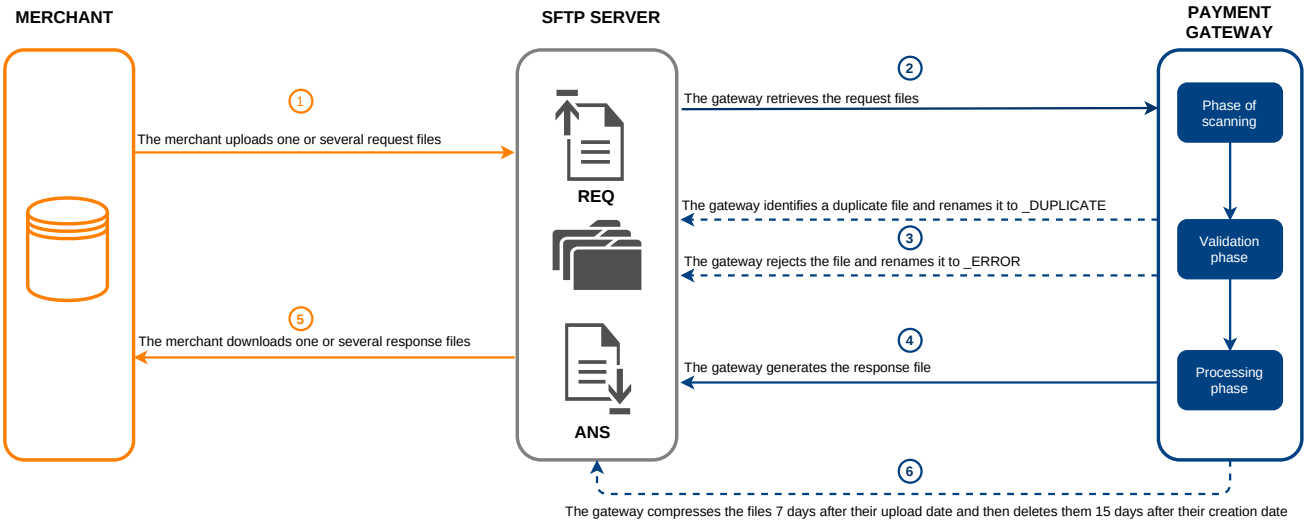
This service makes it possible to carry out these operations in the form of “batch processing”: the merchant site sends a series of orders to the payment gateway in the form of files. The payment gateway processes these orders and, in turn, generates report files.

3.2. Operating principle

The used file transfer protocol is SFTP.

In SFTP transactions, the payment gateway plays the role of the server, and the merchant website acts 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 SFTP server or generated by the payment gateway are automatically compressed 7 days after their upload date.

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 SFTP server in order to transmit them to the payment gateway for validation.

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.**, Europe/Paris time zone.

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 purged automatically.

Processing phase

When a valid file is processed, the request 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. FORMAT OF EXCHANGED FILES

Each query and response file consists of a series of records.

A “record” is a list of parameters separated by the “;” character ending with a carriage return.

The files are structured as follows:

- A header record containing transmission details.
- A variable number of records, each corresponding to one transaction.
- A trailer record that allows to verify the consistency of the transmitted file.

4.1. Format of request file

Header record

The **entête** record provides information about file characteristics for its identification. It is populated as follows:

Pos.	Description	Values
01	Record code	**00**
02	File type	**PAY**
03	File version	**02**
04	Shop ID	**12345678**
05	Mode (TEST or PRODUCTION)	**TEST**
06	Creation date	In YYYYMMDD format
07	Time of creation	In HHMMSS format
08	Requested execution date.	Reserved for future use. Do not populate

Detail record

IMPORTANT
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.

Pos.	Description
01	Description: Record code. Type: NUMERIC | Length: 2 Value: **02**
02	Description: Sequence number of the detail record in the file. Type: NUMERIC | Length: 6 E.g.: • **1** for the first line of the detail record line. • **2** for the second detail record line.
03	Description: Transaction date in the YYYYMMDD format. Type: NUMERIC | Length: 8 E.g.: 20200603
04	Description: Transaction time in the HHMMSS format. Type: NUMERIC | Length: 6 E.g.: 120615

Pos.	Description
05	Description: Transaction identifier. Type:NUMERIC | Length: 6 Hosted Payment Page equivalent: **vads_trans_id** REST API equivalent: N/A E.g.: 000001
06	Description: Transaction type Type: VARCHAR | Length: 2 Value: **CD**
07	Description: Transaction amount expressed in the smallest currency unit . Type:NUMERIC | Length: 13 Hosted Payment Page equivalent: **vads_amount** REST API equivalent: **amount** E.g.: 3000 for 30,00 EUR
08	Description: Numeric code of the currency (ISO 4217 standard). Type: NUMERIC | Length: 3 Hosted Payment Page equivalent: **vads_currency** REST API equivalent: **currency** E.g.: 978 for euro (EUR) (*See table **vads_currency** in annexes*)
09	Description: Requested capture date in the YYYYMMDD format. Leave empty to indicate the current date. Type: NUMERIC | Length: 8 Hosted Payment Page equivalent: **N/A** REST API equivalent: **N/A** E.g.: 20200603
10	Description: Validation mode. Type: NUMERIC | Length: 1 Hosted Payment Page equivalent: **vads_validation_mode** REST API equivalent: **cardOptions.manualValidation** Possible values: • 0: for automatic validation • 1: for manual validation • Empty: for using the default validation mode. E.g.: 0
11	Description: Alias or to debit. Type: VARCHAR | Length: 50 Hosted Payment Page equivalent: **vads_identifier** REST API equivalent: **paymentMethodToken** E.g.: 59ecb199110145338c5704505760ec31
12	Description: The Merchant ID to be used. Leave empty for using the default contract. Type: VARCHAR | Length: 128 Hosted Payment Page equivalent: **vads_contracts** REST API equivalent: **cardOptions.mid** E.g.: 1234567
13	Description: Order reference. Type: VARCHAR | Length: 32 The only accepted special character is "-". Hosted Payment Page equivalent: **vads_order_id** REST API equivalent: **orderId** E.g.: CX-1254
14	Description: Order details 1. Type: VARCHAR | Length: 255 Hosted Payment Page equivalent: **vads_order_info** REST API equivalent: **metadata.orderInfo** E.g.: Door code 3125
15	Description: Order details 2. Type: VARCHAR | Length: 255 Hosted Payment Page equivalent: **vads_order_info2** REST API equivalent: **metadata.orderInfo2** E.g.: No elevator

Pos.	Description
16	Description: Order details 3. Type: VARCHAR | Length: 255 Hosted Payment Page equivalent: **vads_order_info3** REST API equivalent: **metadata.orderInfo3** E.g.: Express

The End record

Pos.	Description	Values
01	Record code	**01**
02	Number of transmitted detail records. **It is recommended not to exceed 2 000 records per file.**	E.g.: 2

File sample

File name: 20200603.12345678.PAY.REQ.T.01

```
00;PAY;02;12345678;TEST;20200603;102008;
02;1;20200603;102008;600001;CD;93599;978;20200604;0;59ecb199110145338c5704505760ec31;;CX-1254;;;
02;2;20200603;102008;600002;CD;7590;978;;0;3d62ec7ce4b249ffb53aa105419aae82;1999888;CX-1255;;;
01;2
```

Naming rules

The names of reporting files should follow strict guidelines and provide several details separated by points.

IMPORTANT
This file type does not have an extension. Make sure you do not add the .txt or csv extensions.

Nomenclature	Description
<date>	Date of file generation in the YYYYMMDD format. E.g.: 20200603
<shopId>	The 8-digit shop identifier as it was defined in the Expert Back Office E.g.: 12345678
PAY	Type of operation (PAYMENT)
REQ	File type (REQUEST).
<mode>	2 possible values: • **T** for a TEST mode file. • **P** for a PRODUCTION mode file.
<sequence>	Sequential number (2 numeric characters) of the file within the same day. E.g.: • 01 for the 1st file of the day, • 02 for the 2nd file of the day, • 99 for the 99th file of the day.

Filename example: 20200603.12345678.PAY.REQ.T.01

4.2. Format of response file

Header record:

Pos.	Description	Values
01	Record code	**00**
02	File type	**PAY**
03	File version	**02**
04	Return code of the first stage of processing	0 : Processing completed 1 : Format error 2 : Value error
05	Information about the error	Additional information about the error, empty if successfully processed.
06	Shop ID	E.g.: 12345678
07	Mode	**TEST** or **PRODUCTION**
08	Creation date in the YYYYMMDD format	E.g.: 20200603
09	Creation time in the HHMMSS format.	E.g.: 145410
10	Date of file generation in the YYYYMMDD format	E.g.: 20200603
11	End of processing time in HHMMSS format	E.g.: 145417

Detail records:

Pos.	Description
01	Description: Record code. Type: NUMERIC | Length: 2 Value: **02**
02	Description: Sequence number of the detail record in the file. Type: NUMERIC | Length: 6 E.g.: • **1** for the first line of the detail record line. • **2** for the second detail record line.
03	Description: Transaction date in the YYYYMMDD format. Type: NUMERIC | Length: 8 E.g.: 20200603
04	Description: Transaction time in the HHMMSS format. Type: NUMERIC | Length: 6 E.g.: 120615
05	Description: Transaction identifier. Type: NUMERIC | Length: 6 Hosted Payment Page equivalent: **vads_trans_id** REST API equivalent: **cardDetails.legacyTransId** E.g.: 700001
06	Description: Transaction type Type: VARCHAR | Length: 2 Value: **CD**
07	Description: Transaction amount expressed in the smallest currency unit . Type: NUMERIC | Length: 12 Hosted Payment Page equivalent: **vads_amount** REST API equivalent: **amount** E.g.: 3000 for 30,00 EUR
08	Description: Numeric code of the currency (ISO 4217 standard). Type: NUMERIC | Length: 3 Hosted Payment Page equivalent: **vads_currency** REST API equivalent: **currency** E.g.: 978 for euro (EUR) (*See table **vads_currency** in annexes*)
09	Description: Transaction amount expressed in the smallest unit of the currency used for the capture in the bank. Type: NUMERIC | Length: 12 Hosted Payment Page equivalent: **vads_effective_amount** REST API equivalent: **transactions.transactionDetails.effectiveAmount** E.g.: 3000 for 30,00 EUR

Pos.	Description
10	Description: Numeric code of the currency that will be used for the capture at the ban (ISO 4217 standard). Type: NUMERIC | Length: 3 Hosted Payment Page equivalent: **vads_effective_currency** REST API equivalent: **transactions.transactionDetails.effectiveCurrency** E.g.: 978 for euro (EUR) (*See table **vads_currency** in annexes*)
11	Description: • Date and time (in the YYYYMMDD format) in UTC time zone of the capture requested by the merchant for a payment by card. • Date and time (in the YYYYMMDD format) in UTC time zone of the funds transfer for a SEPA payment. Type: NUMERIC | Length: 8 Hosted Payment Page equivalent: **vads_presentation_date** REST API equivalent: **transactionDetails.cardDetails.expectedCaptureDate** E.g.: 20200604
12	Description: Validation mode. Type: NUMERIC | Length: 1 Hosted Payment Page equivalent: **vads_validation_mode** REST API equivalent: **cardOptions.manualValidation** Possible values: • 0: for automatic validation • 1: for manual validation E.g.: 0
13	Description: Alias or to debit. Type: VARCHAR | Length: 50 Hosted Payment Page equivalent: **vads_identifier** REST API equivalent: **paymentMethodToken** E.g.: 59ecb199110145338c5704505760ec31
14	Description: MID used to process this request. Type: VARCHAR | Length: 255 Hosted Payment Page equivalent: **vads_contract_used** REST API equivalent: **cardOptions.mid** E.g.: 1234567
15	Description: Order reference. Type: VARCHAR | Length: 32 The only accepted special character is "-". Hosted Payment Page equivalent: **vads_order_id** REST API equivalent: **orderId** E.g.: CX-1254
16	Description: Order details 1. Type: VARCHAR | Length: 255 Hosted Payment Page equivalent: **vads_order_info** REST API equivalent: **metadata.orderInfo** E.g.: Door code 3125
17	Description: Order details 2. Type: VARCHAR | Length: 255 Hosted Payment Page equivalent: **vads_order_info2** REST API equivalent: **metadata.orderInfo2** E.g.: No elevator
18	Description: Order details 3. Type: VARCHAR | Length: 255 Hosted Payment Page equivalent: **vads_order_info3** REST API equivalent: **metadata.orderInfo3** E.g.: Express
19	Description: Return code of processing. Type: NUMERIC | Length: 2 Values: • **00** : Successfully processed – Payment accepted

Pos.	Description
	• **05** : Successfully processed – Payment rejected • **30** : Parameter error. Additional information is provided in field 25. • **96** : Technical error Additional information is provided in field 25.
20	Description: Code sent by the bank that issued the authorization request Type: NUMERIC | Length: 2 Hosted Payment Page equivalent: **vads_auth_result** REST API equivalent: **transactionDetails.cardDetails.authorizationResponse.authorizationResult** Values: See chapter **vads_auth_result**. E.g.: 00
21	Description: Transaction authorization number. Type: VARCHAR | Length: 6 Hosted Payment Page equivalent: **vads_auth_number** REST API equivalent: **transactionDetails.cardDetails.authorizationResponse.authorizationNumber** E.g.: 3fc116
22	Description: Authorization mode. Type: VARCHAR | Length: 4 Values: • **FULL**: Payment authorized. • **MARK**: Payment waiting for authorization.
23	Description: Authorization date in YYYYMMDD format, written in the UTC timezone. Type: DATE | Length: 8 Hosted Payment Page equivalent: N/A REST API equivalent: **transactionDetails.cardDetails.authorizationResponse.authorizationDate** E.g.: 20200114
24	Description: Authorization date in YYYYMMDD format, written in the UTC timezone Type: TIME | Length: 6 Hosted Payment Page equivalent: N/A REST API equivalent: **transactionDetails.cardDetails.authorizationResponse.authorizationDate** E.g.: 142512
25	Description: Additional return code Type: NUMERIC | Length: 2 Description: • Risk assessment result (specific to shops that have enabled the module). The field is empty if no verifications have been completed. • Populated with the position of the error field if the processing code (position 19) is set to 30. • Populated with an error message field if the processing code (position 19) is set to 96. Hosted Payment Page equivalent: **vads_extra_result** REST API equivalent: N/A Values: See chapter vads_auth_result E.g.: 00
26	Description: Payment method number Type: VARCHAR | Length: 36 Hosted Payment Page equivalent: **vads_card_number** REST API equivalent: **transactionDetails.cardDetails.pan** E.g.: 497010XXXXXX0014
27	Description: Exp. date of the payment method in the YYYYMMDD format. Type: NUMERIC | Length: 8 Hosted Payment Page equivalent: **vads_expiry_year** + **vads_expiry_month** REST API equivalent: **transactionDetails.cardDetails.expiryYear** + **transactionDetails.cardDetails.expiryMonth** E.g.: 20211130

End 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	

File sample

```
00;PAY;02;0;;12345678;TEST;20200603;102008;20200603;171049
02;1;20200603;102008;600001;CD;93599;978;93599;978;20200604;0;59ecb199110145338c5704505760ec31;
1234567;CX-1254;;;;00;;;;;;00;;20230603
02;2;20200603;102008;600002;CD;7590;978;7590;978;20200603;0;3d62ec7ce4b249ffb53aa105419aae82;
1999888;CMD20200603-1112;info1;info2;info3;00;0;3fd85e5;FULL;20200603;171049;00;
497010XXXXXX0014;20211130
01;2;2;0
```

Naming rules

Nomenclature	Description
<date>	Date of file generation in the YYYYMMDD format. E.g.: 20200603
<shopId>	The 8-digit shop identifier as it was defined in the Expert Back Office E.g.: 12345678
PAY	Type of operation (PAYMENT)
ANS	File type (ANSWER).
<mode>	2 possible values: • **T** for a TEST mode file. • **P** for a PRODUCTION mode file.
<séquence>	Sequential number (2 numeric characters) of the file within the same day. E.g.: • 01 for the 1st file of the day, • 02 for the 2nd file of the day, • 99 for the 99th file of the day,

Filename example: 20200603.12345678.PAY.ANS.T.01

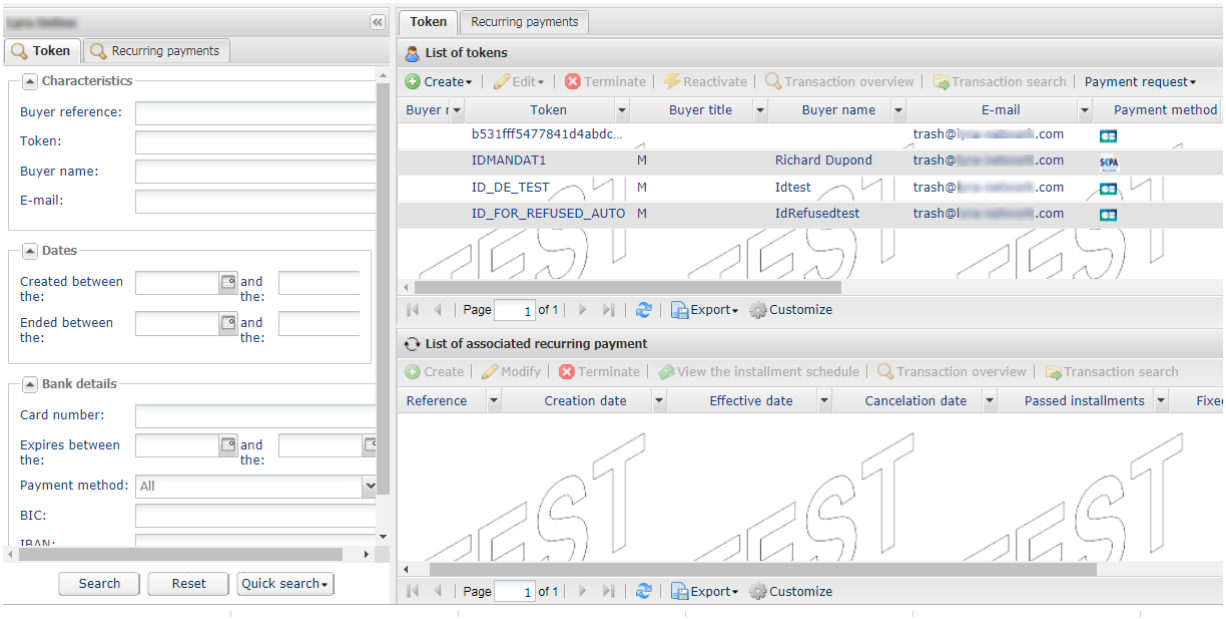
5. VALIDATION PROCESS OF A TEST FILE

Files uploaded to the SFTP server are processed twice a day.

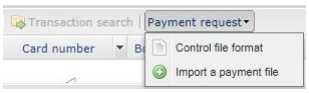
In order not to slow down the implementation of the solution for payment by token in file mode, the merchant can perform **on-demand** tests of their payment files **in TEST mode** in the Expert Back Office.

1. In your Expert Back Office, go to the following menu: **Management** > **TEST recurring payments**

   The TEST recurring payments page appears.



2. Click **Payment request**



Two actions are available:

- Check the file format.
- Import a payment file.

5.1. 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.

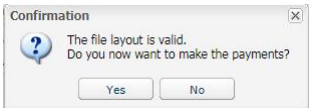
1. Click **Browse**.
2. Select the file to import.

   The size of the file must not exceed 1kb.

   *The file must be named according to the rule specified in the chapter **Naming rules on page 9**. Example: 20161229.91335531.PAY.REQ.T.01*

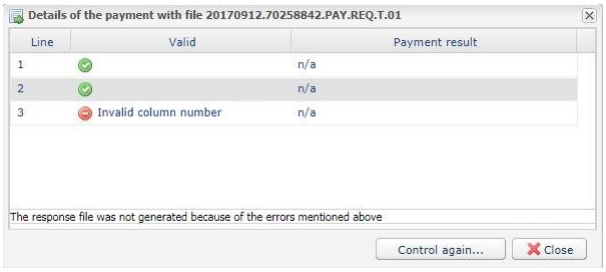
3. Click **Check**.

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** errors

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 file containing errors	Your file does not match the naming convention (see chapter **Naming rules on page 9**).

- 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.

Error message	Description
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 for: • the shop ID, • the environment, • the creation date. Example: Filename: ***20161229.91335531****.PAY.REQ.**T**.01* Header: `00;PAY;02;91335531;TEST;20161229;140800;`

- 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**.
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: • 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.

5.2. 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.

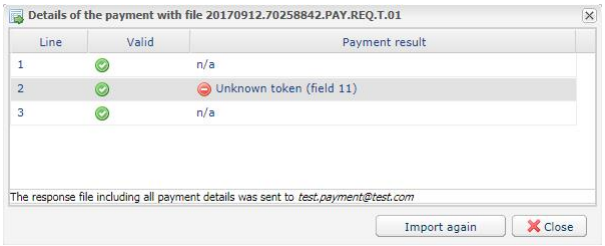
   The size of the file must not exceed 1kb.

   *The file must be named according to the rule specified in the chapter **Naming rules on page 9**. 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 lust have the YYYYMMDD format.
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

Error message	Description
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 must have the YYYYMMDD format.
VALIDATION_MODE format error	Possible values: • 0 (automatic validation), • 1 (manual validation), • empty (default configuration as defined in the Expert 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: • 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.

5.3. Analyzing the 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 maximum size of the file sent is 1kb. It is named according to the rule defined in the chapter **Naming rules on page 13**.

Example: 20161229.91335531.PAY.ANS.T.01

The payment result is displayed in the **Transactions is progress** tab among the test transactions in your Expert Back Office.

You can view the transaction details of the duplicate transaction by clicking on it.

6. UPLOADING A FILE TO THE SFTP SERVER

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.**, Europe/Paris time zone.

It is recommended to upload the files to the SFTP 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 SFTP server:
   - Host: *sftp://vadftp.lyra-network.com*
   - port: 222

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

2. Upload your file into the **shopId/depot** directory.

7. PROVIDING THE RESPONSE FILE

The response file will be available on the SFTP server between 8 a.m. and 9 a.m. and between 2 p.m. and 3 p.m., Europe/Paris time zone.

If processing was completed without technical problems, a response file is available in the **shopId/id** directory.

The query file is automatically moved to the same directory.

The files in the **shopId/id** directory are archived and then automatically purged. You cannot delete them.

In case of a file validation error, the request file is renamed with a **_ERROR** or **_DUPLICATE** suffix in the **shopId/depot** directory.

Once the error has been identified and fixed you can remove these files from **shopId/depot** directory.

Otherwise, they will be automatically deleted after 7 days.

8. ERROR HANDLING

File validation error

Error cases	Cause of the error
The “REQ” file is always present in the /depot directory.	A technical incident prevented the files from being processed. Contact the tech support.
The “ANS” file is absent from the /id directory.	Your REQ file has been rejected. Check for a _ERROR or _DUPLICATE file in the /depot directory.
Presence of an _ERROR file.	• The file name does not respect the naming rule. • The file is empty (its size is 0 bytes).
Presence of a _DUPLICATE file.	A file with the same name has already been processed.

Errors while processing the file, visible in the response file

Return code (position 19)	Additional return code (position 25)	Cause of the error
30	1	The record code must be set to **02**.
30	2	The format of the record sequence number is incorrect.
30	3	The format of the payment date is incorrect.
30	4	The format of the payment time is incorrect.
30	5	The format of the transaction ID is incorrect.
30	6	The transaction type must be populated with **CD**.
30	7	The amount format is incorrect.
30	8	The currency is unknown or its format is incorrect.
30	13	The format of the order reference is incorrect.
96	contratAccepteur.nomatch.cardtype.notaccepted	The contract number defined in position 12 in the query file does not support the type of payment method corresponding to the alias/token defined in position 11.
96	transaction.exist	The transaction number defined in position 5 in the query file already exists.

9. NOTIFICATIONS

9.1. 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/her information system.

9.2. Confirmation e-mail of payment

No confirmation e-mail will be sent by the payment gateway to the buyer or to the merchant in case of accepted payment.

No confirmation e-mail will be sent by the payment gateway to the merchant in case of declined payment.

10. 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 *Hosted Payment Page Implementation Guide* available in our online document archive.

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

Codes returned on the **CB** network:

Value	Description	Grounds of fraud	Value	Description	Grounds of fraud
00	Approved or successfully processed transaction		**43**	Stolen card	YES
02	Contact the card issuer		**51**	Insufficient balance or exceeded credit limit	
03	Invalid acceptor	YES	**54**	Expired card	YES
04	Keep the card	YES	**55**	Incorrect secret code	
05	Do not honor	YES	**56**	Card absent from the file	YES
07	Keep the card, special conditions	YES	**57**	Transaction not allowed for this cardholder	YES
08	Confirm after identification		**58**	Transaction not allowed for this cardholder	
12	Incorrect Transaction Code	YES	**59**	Suspected fraud	YES
13	Incorrect Transaction Amount	YES	**60**	The acceptor of the card must contact the acquirer	
14	Invalid cardholder number	YES	**61**	Withdrawal limit exceeded	
15	Unknown issuer	YES	**63**	Security rules unfulfilled	YES
17	Canceled by the buyer		**68**	Response not received or received too late	
19	Retry later		**75**	Number of attempts for entering the secret code has been exceeded	
20	Incorrect response (error on the domain server)		**76**	The cardholder is already blocked, the previous record has been saved	YES
24	Unsupported file update		**80**	Contactless payment is not accepted by the issuer	YES
25	Unable to locate the registered elements in the file		**81**	Unsecured payment is not accepted by the issuer	YES
26	Duplicate registration, the previous record has been replaced		**82**	Revocation of recurring payment for the card of a specific Merchant or for the MCC and the card	YES
27	File update edit error		**83**	Revocation of all recurring payments for the card	YES
28	Denied access to file		**90**	Temporary shutdown	
29	Unable to update		**91**	Unable to reach the card issuer	
30	Format error		**94**	Duplicate transaction	
31	Unknown acquirer company ID	YES	**96**	System malfunction	
33	Expired card	YES	**97**	Overall monitoring timeout.	

Value	Description	Grounds of fraud	Value	Description	Grounds of fraud
34	Suspected fraud	YES	**98**	Server not available, new network route requested	
38	Expired card		**99**	Initiator domain incident	
41	Lost card	YES			

Codes returned by **Amex Global** acquirer:

Code	Description
000	Approved
001	Approved with an ID
002	Partial approval (Prepaid Cards only)
100	Declined
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
900	Accepted - ATC Synchronization
909	System malfunction (cryptographic error)
912	Issuer not available

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_currency

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

To use a currency other than euro (978), you must request the activation of the "currency conversion" option.

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
Canadian Dollar (CAD)	124	2

Currency	ISO 4217 encoding	Number of digits after the decimal point
Danish Crown (DKK)	208	2
Japanese Yen (JPY)	392	0
Norwegian Crown (NOK)	578	2
Pound Sterling (GBP)	826	2
US Dollar (USD)	840	2
Euro (EUR)	978	2
Polish Zloty (PLN)	985	2

Category Transaction 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.

- 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 on the Merchant's greylist.
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 information.

