

Upload/download files with Ingenico ePayments Batch (advanced)

Table of contents

1. Introduction

2. File format: Upload

2.1 General

2.2 File fields

2.2.1 Layout

2.2.2 Minimum required fields for new transactions

2.2.3 Magstripe/swipe transactions

2.3 Headers and Footer

2.3.1 OHL: Login information header

2.3.2 OHF: File information header

3. Maintenance operations

3.1 File specifics

3.1.1 OHF Header

3.1.2 Transaction level

4. Automatic upload process

4.1 Request URL and specific parameters

4.2 Upload process

4.2.1 Send

4.2.2 Check

4.2.3 Process

4.3 XML response tags

4.4 Security

4.4.1 Secure requests

4.4.2 IP address

5. Automatic download process

5.1 Request URL and parameters

5.2 Format: Electronic reporting

5.2.1 File structure

5.2.2 File format

5.2.3 Extra features

6. Automatic upload and download test

7. File statuses

8. Payment method specifics

8.1 Direct Debits

8.1.1 Direct Debits AT

8.1.2 Direct Debits DE (ELV)

8.1.3 Direct Debits NL

8.2 Payment methods with only maintenance via Batch

8.3 Maestro/MasterCard Advanced Registration Program

9. Other file formats

9.1 Purchasing cards (e-Supply)

9.2 Others

1. Introduction

This document explains the advanced (automatic) integration procedure for Batch mode.

Batch mode allows you to group your payments into formatted files that can be used for uploading and downloading payment results. This option is specially suited to large volumes of payments that do not need online processing or recurrent invoicing.

The file format described in this documentation is the standard Ingenico ePayments file format. For other file formats, go to [Other file formats](#).

For more information on the basic (manual) integration procedure for Batch mode, please refer to the [Basic Batch documentation](#).

2. File format: Upload

The following section contains the standard transaction file formatting rules. If you are using another specific format compatible with our system, go to [Other file formats](#).

2.1 General

A payment file should be formatted in compliance with the following few basic rules:

- The file should be an ASCII text file
- One line per order. The lines are separated by the Carriage Return and Line Feed characters (ASCII : 13 10 – HEX : 0xD 0xA)
- The fields must be separated by a semicolon (“;”)
- The fields themselves cannot contain any semicolons (“;”)

For new transactions (ATR), we automatically split files that are larger than 2,750 lines/transactions. If this is not yet the case for your account, please ask our Customer Care department to enable this functionality for you.

For maintenance transactions (MTR), we don't split files, so the amount of lines should be limited to 30,000. However, to ensure that your files are processed swiftly, we recommend you limit each file to 2,500 lines.

2.2 File fields

2.2.1 Layout

A standard batch file has the following transaction layout:

Field no.	Field	Description
1	AMOUNT	Amount to be paid MULTIPLIED BY 100 since the format of the amount must not contain any decimals or other separators.
2	CURRENCY	ISO alpha order currency code, for example: EUR, USD, GBP, CHF, etc. Note: To ensure optimal processing of your batch files, all currencies configured in your Ingenico ePayments account under Configuration > Account > Currency should be configured accordingly per payment method used (Configuration > Payment methods > Contract data).
3	BRAND	Card brand (See Appendix 3 for non-credit card payment methods).
4	CARDNO	Card/account number.
5	ED	Expiry date (MM/YY or MMY).
6	ORDERID	Your unique order number (merchant reference).
7	COM	Order description.
8	CN	Customer name.
9	PAYID	Our system's unique transaction reference.
10	OPERATION	The payment procedure you have configured in the "Global transaction parameters" tab, "Default

Upload/download files with Ingenico ePayments Batch (advanced)

Field no.	Field	Description
		<p>operation code" section of the Technical Information page will define your default transaction operation. When you send an operation value in the Operation field in your batch, this will overwrite the default value.</p> <p>Possible values for new orders:</p> <ul style="list-style-type: none"> • RES: request for authorisation • SAL: request for direct sale (payment) • RFD: refund, not linked to a previous payment, i.e. not a maintenance operation on an existing transaction (you cannot use this operation without specific permission from your acquirer). <p>Optional:</p> <ul style="list-style-type: none"> • PAU: Request for pre-authorisation In agreement with your acquirer you can use this operation code to temporarily reserve funds on a customer's card. This is a common practice in the travel and rental industry. PAU/pre-authorisation can currently only be used on MasterCard transactions and is supported by selected acquirers. This operation code cannot be set as the default in your Ingenico ePayments account. Should you use PAU on transactions via acquirers or with card brands that don't support pre-authorisation, these transactions will not be blocked but processed as normal (RES) authorisations. <p>Please note that refunds submitted via Batch are not processed until the following day (after the acquirer cut-off time set in your account).</p> <p>See here for the possible values for maintenance operations.</p>
11	AUTHORIZATION CODE	Authorisation code, not received via our system.
12	AUTHORIZATION MODE	The way in which the authorisation code in field 11 was received. Possible value: 'TEL' for telephone
13	AUTHORIZATION DATE	The date/time when the authorisation code in field 11 was received. (MM/DD/YY hh:mm:ss)
14	PSPID	Your affiliation name in our system.
15	GLOBORDERID	Reference grouping several orders together, allows you to request a maintenance operation on all these transactions at a later time.
...	Blank fields	...
22	OWNERADDRESS	Customer's street name and number.
23	OWNERZIP	Customer's postcode.
24	OWNERTOWN	Customer's town/city name.
25	OWNERCTY	Customer's country.

Field no.	Field	Description
26	OWNERTELNO	Customer's telephone number.
27	CVC	Card Verification Code (Card Verification Value).
...	Blank fields	
35	ECI	<p>Electronic Commerce Indicator. You can configure a default ECI value in the "Global transaction parameters" tab, "Default ECI value" section of the Technical Information page. When you send an ECI value in the ECI field in your batch, this will overwrite the default ECI value.</p> <p>0 - Swiped</p> <p>1 - Manually keyed (MOTO, card not present)</p> <p>2 - Recurring payments, originating from MOTO</p> <p>3 - Installment payments</p> <p>7 - E-commerce with SSL encryption</p> <p>9 - Recurring after first e-commerce transaction</p>

Additional fields may also be optionally sent by merchants who have activated certain options/functionalities in their accounts. Please refer to the respective option documentation for more information on additional fields linked to the option.

2.2.2 Minimum required fields for new transactions

This section only applies to new transactions. Go to [Maintenance operations](#) for more information on maintenance operations on existing transactions.

General: new transaction

For **new transactions** that you would like to enter into our system, the minimum fields required at the transaction level are:

- Amount (field 1 in the file layout)
- Currency (field 2)
- Card (or account) number (field 4)
- Expiry date (field 5)
- CVC (field 27) (this field is required depending on your acquirer and the ECI value for your transactions)
- Your order reference (field 6)
- Operation (field 10)

Example

General: new transaction

```
10000;EUR;;4111111111111111;11/12;Order0001;;Paul Smith;;SAL;;;;;;;;;;;;;;;123;
```

```
9500;EUR;;5399999999999999;11/10;Order0002;;Jane Doe;;SAL;;;;;;;;;;;;;;;485;
```

Upload/download files with Ingenico ePayments Batch (advanced)

```
2050;EUR;;4444333322221111;11/09;Order0003;;John Doe;;SAL;;;;;;;;875;
```

Exception: authorisations received via an alternative channel

When a (related) transaction does not yet exist in our system, the minimum fields required at the transaction level for authorisations received via an alternative channel (e.g. an authorisation received by phone with your acquirer) are:

- Amount (field 1 in the file layout).
- Currency (field 2).
- Card (or account) number (field 4).
- Expiry date (field 5).
- Your order reference (field 6).
- Card (account) holder name (field 8).
- Operation (field 10).
- Authorisation code (field 11).
- Authorisation mode (field 12).
- Authorisation date (field 13).

Example

Exception: authorisation already received through an alternative channel

```
1000;EUR;;4111111111111111;11/12;Order0001;;Paul Smith;;IMP;43785;TEL;08/08/07 15:15:52;
```

```
9500;EUR;;5399999999999999;11/10;Order0002;;Jane Doe;;IMP;83145;TEL;08/08/07 15:21:45;
```

```
2050;EUR;;4111111111111111;11/09;Order0003;;John Doe;;IMP;73586;TEL;08/08/07 15:26:12;
```

2.2.3 Magstripe/swipe transactions

For transactions accepted inflight through magnetic stripe/swipe, "TRACK2" data can be sent in line 42 of a batch file.

TRACK2 is a track read by ATMs and credit card checkers. It contains the cardholder's account, encrypted PIN, and other discretionary information.

2.3 Headers and Footer

You will need to use the headers and the footer if you are [uploading your files automatically](#) or sending maintenance operations in your file.

There are two headers: **OHL** containing login information and **OHF** containing general file information. The headers have to be entered in the first lines of the file, before the actual payment data.

There is one general file footer called **OTF**. The file footer has to be entered as the last line of the file, after the actual payment data. The footer layout is simply "OTF;".

The following example shows a file containing both headers and footer. In the following sections, we will take a closer look at the layout of the login information and file information headers.

Example

```
OHL;FishShop1;MySecret84;;FishAPI;
```

```
OHF;File53484;ATR;RES;2;

10000;EUR;;4111111111111111;11/12;Order0001;golf balls;Paul Smith;;RES;;123;
9500;EUR;;5399999999999999;11/10;Order0002;mobile phone;Jane Doe;;RES;;485;

OTF;
```

2.3.1 OHL: Login information header

Only use a Login Header when you are uploading your files automatically (exception: manual file upload on a Group).

Login Header layout:

OHL;PSPID;PSWD;USERTYPE;USERID;

Field	Description
OHL	Fixed value, indicating this is a header containing login information
PSPID	Your affiliation name in our system
PSWD	The password belonging to your user (> USERID)
USERTYPE	Only applicable for Group structures, value: MGID
USERID	Your (API) user (please refer to the User Manager documentation for more information on API users)

The fields you fill in will depend on whether you use the login details from:

- Your PSPID and a USERID
- A Group: when you have a “Group” structure and you want to send a file containing transactions for more than one PSPID belonging to your group, the login details need to contain a (default) PSPID that belongs to your group and a “Group” level user (as opposed to a user belonging to a specific PSPID).

We will illustrate this in the following examples:

Examples

Merchant login header

Mike has a PSPID called MilkyMike. He wants to submit his files automatically from an application at his end. To do this, he has created an API user called MikeAPI in his account, with A78H29U41 for the password. His login header would be:

```
OHL;MilkyMike;A78H29U41;;MikeAPI;
```

Group login header

John Doe Trading has 3 different PSPIDs: JohnUK, JohnUS and JohnAU. They have a group structure, called JohnDGroup, grouping the 3 PSPIDs together. In JohnDGroup, they have created an API user called JohnUser1 with MySecret12 for the password.

John wants to automatically submit a file containing transactions for the 3 different PSPIDs. He has the most transactions in the UK. This is what his login header will look like:

Upload/download files with Ingenico ePayments Batch (advanced)

```
OHL;JohnUK;MySecret12;MGID;JohnUser1;
```

2.3.2 OHF: File information header

File Header layout:

```
OHF;FILE_REFERENCE;TRANSACTION_CODE;OPERATION;NB_PAYMENTS;
```

Field	Description
OHF	Fixed value, indicating this is a header containing general file information
FILE_REFERENCE	Mandatory reference (name) for the file, freely definable. You can use this reference to look up your file afterwards. (max. 50 characters)
TRANSACTION_CODE	Possible values: <ul style="list-style-type: none">• ATR: code for new orders (transactions)• MTR: code for maintenance operations on existing transactions Any file you upload is automatically an ATR file unless you specify otherwise in the file header.
OPERATION	For new orders: left blank, RES, SAL, RFD For maintenance operations: REN, DEL, DES, SAL, SAS, RFD, RFS (Go to File fields and File specifics for operation explanations) The OPERATION parameter at the file level can be overwritten by the OPERATION parameter at transaction level.
NB_PAYMENTS	Number of payments in the file. Numeric. Mandatory unless you are using a file footer (in which case we still suggest you use this field).

Example

File information header for new orders

Jane has a file called Membership0607 containing 86 requests for authorisation. Her file information header would be:

```
OHF;Membership0607;ATR;RES;86;
```

File information header for maintenance transactions

Mike has a file called RefundsJune containing 9 (final) refunds. His file information header would be:

```
OHF;RefundsJune;MTR;RFS;9;
```

The login and file information details can be transmitted in file headers, as illustrated above, or as parameters in HTTP POST requests (only for [automatic file upload](#)).

Upload/download files with Ingenico ePayments Batch (advanced)

3. Maintenance operations

Maintenance operations are operations performed on already authorised or paid orders, in other words on transactions already existing in our system.

To perform certain maintenance operations, they have to be enabled for your account (sometimes the availability depends on your subscription), your acquirer must allow the operation, and the operation must be possible for the payment method concerned.

3.1 File specifics

3.1.1 OHF Header

The TRANSACTION_CODE for maintenance operations is **MTR**.

The OPERATION code for the maintenance operation is provided at a file and/or individual transaction level. The operation code at the transaction level will overwrite the operation code at the file level for maintenance operations. This means you can send one file containing different operations, e.g. partial and last data captures, refunds, etc.

The possible values for maintenance operations (field 10) are:

- **DEL**: delete authorisation, leaving the transaction open for further potential maintenance operations.
- **DES**: delete authorisation, closing the transaction after this operation.
- **SAL**: partial data capture (payment), leaving the transaction open for another potential data capture.
- **SAS**: (last) partial or full data capture (payment), closing the transaction for further data captures.
- **RFD**: partial refund (on a paid order), leaving the transaction open for another potential refund.
- **RFS**: (last) partial or full refund (on a paid order), closing the transaction after this refund.

In general you will perform the operations **REN**, **DEL**, **DES**, **SAL**, **SAS** on orders in status 5-Authorised and the operations **RFD**, **RFS** on orders in status 9-Payment requested.

3.1.2 Transaction level

The fields required at a transaction level for a manual authorisation after the initial authorisation (via our system) was refused (order in status 2-Authorisation refused) are:

- Amount (field 1 in the file layout)
- Currency (field 2)
- Our system's PAYID (field 9)
- Operation code (field 10)
- Authorisation code (field 11)
- Authorisation mode (field 12)
- Authorisation date (field 13).

Example

Mike has a PSPID called MilkyMike. He wants to submit his files automatically from an application at his end. To do this, he has created an API user, called MikeAPI, in his account, with A78H29U41 as the password. His login header would be:

```
OHL;MilkyMike;A78H29U41;;MikeAPI;
```

1. Authorisation

Mike has a file called AutoMay, containing 1 transaction: an authorisation request for 75 EUR. His file information header would be:

OHF;AutoMay;ATR;RES;1;

This is the file he submits:

OHL;MilkyMike;A78H29U41;;MikeAPI;

OHF;AutoMay;ATR;RES;1;

7500;EUR;;5399999999999999;11/10;Order0001;;Jane Doe;;RES;;;;;;;;;;;;;485;

OTF;

After the full upload process, the file is assigned FileID 1543 and the transaction is assigned the following reference from our system (PAYID): 1348645

In the account back office, there is now a PAYID 1348645 with a historic record in status 5-Authorised.

2. Partial data capture

Mike wants to capture data for a part of the authorised amount, leaving the transaction open for a final data capture for the rest of the amount later (operation code: SAL). He wants to capture an amount of 25 EUR for PAYID 1348645. His file is called DataCap1May. His file information header would be:

OHF;DataCap1May;MTR;SAL;1;

This is the file he submits:

OHL;MilkyMike;A78H29U41;;MikeAPI;

OHF;DataCap1May;MTR;SAL;1;

2500;EUR;;;;;;;;;1348645;SAL;

OTF;

After the full upload process, the file is assigned FileID 1571. In the account back office, PAYID 1348645 now has 2 historic records: /0 in status 5-Authorised for an amount of 75 EUR and /1 in status 9-Payment requested for an amount of 25 EUR. The order status is still 5-Authorised because the submitted data capture was partial.

3. Final data capture

Mike wants to make a LAST data capture for the rest of the authorised amount (operation code: SAS), i.e. a data capture for an amount of 50 EUR for PAYID 1348645. His file is called DataCap2May. His file information header would be:

OHF;DataCap2May;MTR;SAS;1;

This is the file he submits:

OHL;MilkyMike;A78H29U41;;MikeAPI;

OHF;DataCap2May;MTR;SAS;1;

5000;EUR;;;;;;;;;1348645;SAS;

OTF;

After the full upload process, the file is assigned FileID 1610. In the account back office, PAYID 1348645 now has 3 historic records: /0 in status 5-Authorized for an amount of 75 EUR, /1 in status 9-Payment requested for an amount of 25 EUR and /2 in status 9-Payment requested for an amount of 50 EUR. The order status is now 9- Payment requested because the submitted data capture was the LAST one (closing the transaction).

4. Refund

Mike wants to refund (operation code: RFS) the customer for the full amount, a refund for an amount of 75 EUR for PAYID 1348645. His file is called RefundMay. His file information header would be:

OHF;RefundMay;MTR;RFS;1;

This is the file he submits:

OHL;MilkyMike;A78H29U41;;MikeAPI;

OHF;RefundMay;MTR;RFS;1;

7500;EUR;;;;;;;;;1348645;RFS;

OTF;

After the full upload process, the file is assigned FileID 1671. In the account back office, PAYID 1348645 now has 4 historic records: /0 in status 5-Authorized for an amount of 75 EUR, /1 in status 9-Payment requested for an amount of 25 EUR, /2 in status 9-Payment requested for an amount of 50 EUR and 3/ in status 8-Refund for an amount of 75 EUR. The order status is now 8-Refund because the submitted refund was the LAST one (closing the transaction).

4. Automatic upload process

There are 2 different ways to upload a file:

1. **Manual file upload:** You can upload the payment files manually from your account's back office. Go to manual Batch for more information.
2. **Automatic upload from a merchant application:** the merchant application makes https file upload and download requests on specific pages in our system. This requires developments in the merchant application.

This chapter describes the automatic upload process from a merchant application.

4.1 Request URL and specific parameters

An Automatic File Upload allows you to process multiple payments directly from your own system (programme, scheduled task, etc.) without the need for a browser or human intervention. Your server sends an HTTPS POST request to our server. The transaction file and additional parameters are transmitted in the "content" of the https request.

Your application sends an HTTPS request to the following page on our server:

- Test: https://ogone.test.v-psp.com/ncol/test/AFU_agree.asp (will be used in further examples)
- Production: https://secure.ogone.com/ncol/prod/AFU_agree.asp

The file format is identical for manual and automatic file uploads. In the event of automatic file upload however, login data, certain process data and general parameters are mandatory in the request. You can submit these parameters as header/footer parameters included in the file, or as POST parameters separately from the file.

The general parameters set out below need to be sent in each automatic file upload request. These parameters refer to "Content-disposition" in the HTTP protocol and can only be specified as POST parameters separately from the file, not in headers.

Field	Description
REPLY_TYPE	<p>Defines which response format is requested from our system. Possible values:</p> <ul style="list-style-type: none"> • XML (default) • HTML <p>The HTML reply contains the same page as when a file is uploaded manually.</p>
PROCESS_MODE	<p>Possible values for the processing mode:</p> <ul style="list-style-type: none"> • SEND: a new file is uploaded but no format check is requested • CHECK: a simple format check is requested (corresponds to step 2 of the manual file upload) • PROCESS: the processing of an already uploaded file is requested (corresponds to step 3 of the manual file upload) • CANCEL: the cancellation of an already checked (but not loaded) file is requested (corresponds to the cancel button after step 2 of the manual file upload) • Some steps can also be combined (not recommended): • CHECK: can combine SEND and CHECK (the transaction file is transmitted with PROCESS_MODE=CHECK but without our system's FileID that is normally returned after the SEND step). • CHECKANDPROCESS: can combine CHECK and PROCESS or SEND, CHECK and PROCESS.
MODE	<p>Can be used to specify whether your application is waiting for a response from our server or if you will retrieve the response later. Valid modes are:</p>

Field	Description
	<ul style="list-style-type: none"> • SYNC (Default) • ASYNC <p>Currently this option only has an effect on the PROCESS or CHECKANDPROCESS steps (see PROCESS_MODE above).</p> <p>In SYNC mode, you will receive the result of the payment validation process if you stay connected (not the result of the authorisation with the acquirer(s), as this always occurs offline for file uploads).</p> <p>In ASYNC mode, the validation process is performed offline and you only receive the FILE ID as confirmation. An email is sent to the merchant when the file has been validated by our system. If you send large files, we advise you to use ASYNC mode.</p> <p>Independently of the mode used, an email is also sent once the file has been processed (submitted to the acquirer).</p>

Examples

Additional parameters specified as separate POST parameters

```
<form action="https://ogone.test.v-psp.com/ncol/test/AFU_agree.asp" method=POST>
<textarea name=FILE>
1000;EUR;;4111111111111111;11/12;Order0001;golf balls;Paul Smith;;;;;;;;;;;;;;;123;
9500;EUR;;5399999999999999;11/10;Order0002;mobile phone;Jane Doe;;;;;;;;;;;;;;;485;
2050;EUR;;4444333322221111;11/09;Order0003;cocktails;John Doe;;;;;;;;;;;;;;;875;
</textarea>
<input type=text name=FILE_REFERENCE value="File53484">
<input type=text name=PSPID value="FishShop1">
<input type=text name=USERID value="FishAPI">
<input type=text name=PSWD value="MySecret81">
<input type=text name=TRANSACTION_CODE value="ATR">
<input type=text name=OPERATION value="RES">
<input type=text name=NB_PAYMENTS value="3">
<input type=text name=REPLY_TYPE value="XML">
<input type=text name=MODE value="SYNC">
<input type=text name=PROCESS_MODE value="SEND">
</form>
```

Additional parameters specified in the file headers/footer

```
<form action="https://ogone.test.v-psp.com/ncol/test/AFU_agree.asp" method=POST>
<textarea name=FILE> OHL;FishShop1;MySecret81;;FishAPI;
OHF;File53484;ATR;RES;3;
1000;EUR;;4111111111111111;11/12;Order0001;golf balls;Paul Smith;;RES;;;;;;;;;;;;;;;123;
9500;EUR;;5399999999999999;11/10;Order0002;mobile phone;Jane Doe;;RES;;;;;;;;;;;;;;;485;
2050;EUR;;4444333322221111;11/09;Order0003;cocktails;John Doe;;RES;;;;;;;;;;;;;;;875;
</textarea>
```

```
<input type=text name=REPLY_TYPE value="XML">
<input type=text name=MODE value="SYNC">
<input type=text name=PROCESS_MODE value="SEND">
</form>
```

4.2 Upload process

We advise you to split the upload process into the different steps: SEND, CHECK and PROCESS ("PROCESS_MODE" values). These steps correspond to the different steps of the manual file upload.

There are two main advantages of splitting the upload process:

- The split between CHECK and PROCESS allows your application to cancel the upload after our system's format check, should there be too many format errors in the file.
- The split between SEND and the rest of the processing prevents you from processing the same file more than once. If your application doesn't receive a response after a certain step for any reason, you can repeat the step with no risk. If you perform the upload in a single step (CHECKANDPROCESS) and a communication problem occurred before your application received the response, it might think that the file has not been uploaded and try uploading it a second time, resulting in duplicate processing of the same file.

The merchant's application triggers each step by making an HTTPS POST request to our server. The HTTPS request can be sent as a standard POST HTTPS request (with the file content in a standard field named "FILE") or as a MULTIPART/FORM-DATA POST HTTPS request (with the file content in a "file" type field named "FILE", <input type="file" name="File1">). For both methods, you need a component capable of sending HTTP requests to a secure server.

In the first step (SEND), the file itself is part of the content of the request. Our system returns a FileID in the XML response to this request. You will use this FileID instead of the file itself in the subsequent upload steps (CHECK and PROCESS).

4.2.1 Send

When you send PROCESS_MODE=SEND, you upload a new transaction file, but no check or processing is performed (this corresponds to Step 1 of a manual file upload).

When our system receives the HTTPS request, it checks that the general parameters are valid. If something is wrong, an error is returned to your application and the process stops without performing any operation.

If the general parameters are valid, our system returns a FILEID and a GUID. This FILEID will be used in subsequent requests instead of the file itself. The GUID can be used in subsequent requests to replace Login/Password.

After this step, the file status is set to "Uploaded" (visible in the back office or with the automatic file download).

Example

Request

```
<form action="https://ogone.test.v-psp.com/ncol/test/AFU_agree.asp" method=POST>
<textarea name=FILE>
10000;EUR;;4111111111111111;11/12;Order0001;golf balls;Paul Smith;;RES;;;;;;;;;;;;;;;123;
9500;EUR;;5399999999999999;11/10;Order0002;mobile phone;Jane Doe;;RES;;;;;;;;;;;;;;;485;
2050;EUR;;4444333322221111;11/09;Order0003;cocktails;John Doe;;RES;;;;;;;;;;;;;;;875;
</textarea>
<input type=text name=FILE_REFERENCE value="File53484">
<input type=text name=PSPID value="FishShop1">
<input type=text name=USERID value="FishAPI">
```

```
<input type=text name=PSWD value="MySecret81">
<input type=text name=TRANSACTION_CODE value="ATR">
<input type=text name=OPERATION value="RES">
<input type=text name=NB_PAYMENTS value="3">
<input type=text name=REPLY_TYPE value="XML">
<input type=text name=MODE value="SYNC">
<input type=text name=PROCESS_MODE value="SEND">
</form>
```

XML reply

```
<?xml version="1.0" ?>
<AFU_REPLY>
<SEND_FILE>
<FILEID>2010</FILEID>
<GUID>8B1F5D65D5C7C5C80551D2AA955F810A45BD1752</GUID>
</SEND_FILE>
</AFU_REPLY>
```

4.2.2 Check

When you send PROCESS_MODE= CHECK, a simple format check is requested (this corresponds to Step 2 of a manual file upload).

You will use the FILEID in the request instead of the file itself.

Our system returns format errors, should any such errors occur.

After this step, the file status is set to **Checked**.

Example

Request

```
<form action="<%URL_TESTENV%>AFU_agree.asp" method=POST>
<input type=text name=PSPID value="FishShop1">
<input type=text name=USERID value="FishAPI">
<input type=text name=PSWD value="MySecret81">
<input type=text name=REPLY_TYPE value="XML">
<input type=text name=MODE value="SYNC">
<input type=text name=PFID value="2010">
<input type=text name=PROCESS_MODE value="SEND">
</form>
```

XML reply

```
<?xml version="1.0" ?>
<AFU_REPLY>
<FORMAT_CHECK>
<FORMAT_CHECK_ERROR>
<ERROR>|Wrong credit card number format|</ERROR>
<LINE>1</LINE>
```

```
<NCERROR>50001002</NCERROR>
<PAYID>0</PAYID>
<ORDERID>Order0001</ORDERID>
<NCSTATUS>5</NCSTATUS>
<STATUS>0</STATUS>
</FORMAT_CHECK_ERROR>
<FILEID>2012</FILEID>
</FORMAT_CHECK>
</AFU_REPLY>
```

4.2.3 Process

When you send `PROCESS_MODE=PROCESS`, the processing of the file is requested (this corresponds to Step 3 of a manual file upload).

You will use the `FILEID` in the request instead of the file itself.

Our system returns loading errors, should any such errors occur.

`OK_PAYMENTS` is the number of transactions correctly loaded. `RANGE_START` and `RANGE_STOP` represent the range of `PAYIDs` for the loaded transactions, as assigned by our system (only when the `transaction_code` is `ATR`).

After this step, the file status is set to **"Being Processed"**.

After the response is sent, the transactions are processed with the acquirers (banks) in offline mode. It is possible to retrieve results in your back office or with the automatic file download.

Example: Request in SYNC mode

Request in SYNC mode

```
<form action="https://ogone.test.v-psp.com/ncol/test/AFU_agree.asp" method=POST>
<input type=text name=PSPID value="FishShop1">
<input type=text name=USERID value="FishAPI">
<input type=text name=PSWD value="MySecret81">
<input type=text name=REPLY_TYPE value="XML">
<input type=text name=MODE value="SYNC">
<input type=text name=PFID value="2010">
<input type=text name=PROCESS_MODE value="PROCESS">
</form>
```

XML reply

```
<?xml version="1.0" ?>
<AFU_REPLY>
<PROCESSING>
<NC_ERROR>
<NCERRORPLUS>||Wrong credit card number format|</NCERRORPLUS>
<LINE>1</LINE>
<NCERROR>50001002</NCERROR>
<PAYID>37267</PAYID>
```

```
<ORDERID>Order0001</ORDERID>
<NCSTATUS>5</NCSTATUS>
<STATUS>0</STATUS>
</NC_ERROR>
<FILEID>2011</FILEID>
<SUMMARY>
<NB_PAYMENTS>3</NB_PAYMENTS>
<OK_PAYMENTS>2</OK_PAYMENTS>
<RANGE_START>37267</RANGE_START>
<RANGE_STOP>37269</RANGE_STOP>
</SUMMARY>
</PROCESSING>
</AFU_REPLY>
```

Example: Request in ASYNC mode

Request in ASYNC mode

```
<form action="https://ogone.test.v-psp.com/ncol/test/AFU_agree.asp" method=POST>
<input type=text name=PSPID value="FishShop1">
<input type=text name=USERID value="FishAPI">
<input type=text name=PSWD value="MySecret81">
<input type=text name=REPLY_TYPE value="XML">
<input type=text name=MODE value="ASYNC">
<input type=text name=PFID value="2010">
<input type=text name=PROCESS_MODE value="PROCESS">
</form>
```

XML reply

```
<?xml version="1.0" ?>
<AFU_REPLY>
<PROCESSING>
<FILEID>2011</FILEID>
<SUMMARY>
<NB_PAYMENTS>3</NB_PAYMENTS>
</SUMMARY>
</PROCESSING>
</AFU_REPLY>
```

4.3 XML response tags

Tag	Description
<AFU_REPLY>	Automatic File Upload Reply
<PARAMS_ERROR>	General parameter errors

Upload/download files with Ingenico ePayments Batch (advanced)

Tag	Description
<PARAM>	Parameter name
<ERROR>	Error description
<FILE_ERROR>	File processing error
<ERROR>	Error description
<FORMAT_CHECK>	Individual transaction format check
<FILEID>	Numeric FileID designated by our system
<GUID>	<p>Alphanumeric Login Identifier for the file designated by our system.</p> <p>This GUID is designated the first time you submit a file (only once per FileID). When you want to process the file or perform another valid operation (or download at file level etc.), you can submit this GUID instead of the login parameters described above. It serves as a form of password, granting access only to this specific file.</p>
<FORMAT_CHECK_ERROR>	Transaction format error
<LINE>	Corresponding line number in the file
<ERROR>	Error description
<NCERROR>	Numeric error code
<PROCESSING>	Loading of the individual transactions
<NC_ERROR>	Load error
<LINE>	Payment line
<PAYID>	Our system's PAYID
<NCSTATUS>	Error status (first digit of NCERROR)
<STATUS>	Status of the PAYID after the transaction has been loaded
<NCERROR>	Numeric error code

Tag	Description
<NCERRORPLUS>	Error description
<SUMMARY>	Summary of the file
<NB_PAYMENTS>	Number of received payments
<OK_PAYMENTS>	Number of correctly formatted payments
<RANGE_START>	First PAYID (for ATR-new orders only)
<RANGE_STOP>	Last PAYID (for ATR-new orders only)
<NB_ALIAS>	Number of Alias operations
<NB_SUBSCRIPTION>	Number of Subscription operations

4.4 Security

4.4.1 Secure requests

The automatic file upload and download functions are built on a robust secure communication protocol.

Batch API is a set of instructions submitted with standard HTTPS Post requests. We only allow the merchant to connect to us in secure HTTPS mode.

There is no need for a client SSL certificate.

4.4.2 IP address

The IP address(es) or IP address range(s) of the servers from which you are sending us requests can be configured in the IP address field of the "Data and origin verification" tab, checks for automatic Batch section of the Technical Information page in your account. We will check the IP address when you perform a request for an automatic file upload or download.

If the IP address from which the request originates has not been defined in the IP address field of the "Data and origin verification" tab, checks for automatic Batch section of the Technical Information page in your account, you will receive the error message "unknown order/1/i". The IP address from which the request was sent will also be displayed in the error message.

5. Automatic download process

You can either download a payment file (transaction results) manually via the back office or download your payment files automatically (through an application).

This chapter describes the automatic file download and the electronic reporting page. Go to [Basic Batch](#) for more information on manual file downloads and consulting files in the back office.

5.1 Request URL and parameters

Your application sends an HTTPS request to the following page on our server: https://ogone.test.v-psp.com/ncol/test/payment_download_ncp.asp

(The "test" field indicates it is a TEST URL. Replace "test" by "prod" when working with our production environment.)

The parameters should be sent using the POST method.

Field	Description / Value
PSPID	Your affiliation name in our system
USERID	Name of your API user
PSWD	Password of your API user
level	Level of payment selection Possible values: <ul style="list-style-type: none"> • ORDERLEVEL (Default) • HISTLEVEL (financial history level) • FILELEVEL
ofd	Order date 'from' day (DD)
ofm	Order date 'from' month (MM)
ofy	Order date 'from' year (YYYY)
otd	Order date 'to' day (DD)
otm	Order date 'to' month (MM)
oty	Order date 'to' year (YYYY)
afd	Payment date 'from' day (DD)
afm	Payment date 'from' month (MM)
afy	Payment date 'from' year (YYYY)
atd	Payment date 'to' day (DD)
atm	Payment date 'to' month (MM)

Upload/download files with Ingenico ePayments Batch (advanced)

Field	Description / Value
aty	Payment date 'to' year (YYYY)
arch	Send the value "arch" with this field if your download will contain transactions older than 45 days.
PM	Payment method
BRAND	Card brand
CARDNO	Card/account number
orderID	Your reference for the order
Company	Client company
facname1	Customer name
PAYID	Our system's reference for the payment
ID	FileID of a previously uploaded file (our system's reference)
status	List of specific statuses (numeric code) separated by ",".
OnlyIfProcessed	Indicates that only files with the "Processed" status should be downloaded. If the file is not yet processed, our system will return the message: "File not yet processed: please retry later ..." (to select: OnlyIfProcessed="1").
stok	Only in combination with "OnlyIfProcessed". Select all accepted transactions (status 5, 6, 7, 74, 85, 8, 84, 85, 9, 94, 95) of a processed file (to select: stok="1").
sterr	Only in combination with "OnlyIfProcessed". Select all rejected transactions (status 2, 63, 73, 83, 93) of a processed file (to select: sterr="1")
st1	Status 0, 1, 51, 71, 81, 91. (To select: st1="1")
st2	Status 2. (To select: st2="1")
st3	Status 5. (To select: st3="1")
st4	Status 9, 94. (To select: st4="1")
st5	"Others" status: all statuses other than the ones in st1, st2, st3, st4 and st6 (to select: st5="1")
st6	Status 7, 8, 87, 84 (to select: st6="1")
<u>Electronic reporting specification</u> (optional):	
Structure	File structure. Possible values: <ul style="list-style-type: none"> • STD (standard) • EXT (extended) • MNG (file management) • DYN (dynamic)

Field	Description / Value
Format	File format. Possible values: <ul style="list-style-type: none"> • XML • FIX (fixed length) • DEL (delimited)
Sep	Separator in the event that the file format is DEL (delimited). Default separator: “;”.
headers	File download headers (to select: headers=“1”)
GR1	First criterion to group payments together. Possible values: <ul style="list-style-type: none"> • LASTACTION • BRAND • ORDERDATE • PFID (uploaded file reference) • PAYMENTMETHOD • STATUS
GR2	Second criterion to group payments together. Possible values: see GR1.
GR3	Third criterion to group payments together. Possible values: see GR1.

Important remarks:

- For the date, you either use ofd, ofm, ofy, otd, otm, oty, to search at the order level (order date) or afd, afm, afy, atd, atm, aty, to search at the history level (payment date). It is mandatory for the date to be sent in your requests, unless the FileID or PAYID is sent.
- For the statuses, you should only use one of the three following options:
 - status
 - stok, sterr in combination with OnlyIfProcessed
 - st1, st2, st3, st4, st5, st6

Example

The merchant wants to retrieve all orders for which authorisation has been declined with an order date between 1st and 15th June 2007.

```
<form action="https://ogone.test.v-psp.com/ncol/test/payment_download_ncp.asp" method=POST>
<input type=text name=PSPID value="FishShop1">
<input type=text name=USERID value="FishAPI">
<input type=text name=PSWD value="MySecret&1">
<input type=text name=level value="ORDERLEVEL">
<input type=text name=ofd value="01">
<input type=text name=ofm value="06">
<input type=text name=ofy value="2007">
<input type=text name=otd value="15">
<input type=text name=otm value="06">
<input type=text name=oty value="2007">
<input type=text name=st2 value="1">
</form>
```

5.2 Format: Electronic reporting

Upload/download files with Ingenico ePayments Batch (advanced)

In the Electronic Reporting page ("Electronic Reporting" link in your back-office menu), you can set the format and structure you want to use for electronic reports such as file downloads (when push-reports are activated in your account, the electronic reporting link will give access to a list of your push-reports.)

The file format can be set per user. If you want to change the electronic reporting settings for an API user, you can do so via the User Management page ("Users" link in the menu > "Edit" button next to the API user > "File format >>>" link).

5.2.1 File structure

The structure defines the type of information available for each transaction. The payment result files can be downloaded in 4 different structures: "Standard", "Extended", "File management" or "Dynamic".

The full description of these structures is available via the "More info" link on the Electronic reporting page. Please refer to the links in this page for the XML tag names, description, size and format of the download fields.

5.2.2 File format

There are three different formats for file downloads: delimited, XML and fixed length. The layout and an example for each of these are provided below. The file structure used in the examples is "Standard".

Delimited

Layout

<DOWNLOAD_REPLY> Beginning of the file

<TRANSMISSION Description string>

Sequential list of payments: each line corresponds to one order. Each field is separated by the separator specified in the "Electronic Reporting" page. Each occurrence of a separator in a field value is replaced by a blank.

<END_TRANSMISSION> End of the payments selection

Example

<DOWNLOAD_REPLY>

<TRANSMISSION Order Level: - Order date 12/04/2007- Status All except Cancelled by client,Invalid or incomplete->

1651818;order0001;12/4/2007;5;Authorised;testoff;;;Paul Smith;;120.00;EUR;0.00;0.00;CreditCard;VISA;XXXXXXXXXXXX1111;;

1651819;order0002;12/4/2007;5;Authorised;testoff;;;Bill Durand;;56.00;EUR;0.00;0.00;CreditCard;Eurocard;XXXXXXXXXXXX9999;;

<END_TRANSMISSION>

XML

Layout

<?xml version="1.0"?>

<DOWNLOAD_REPLY> Beginning of the file

In the event of an error:

<SYSTEM_ERROR>: System error message

<PARAMS_ERROR>: Parameters error message

<NCERROR>: Numeric error code

or

Upload/download files with Ingenico ePayments Batch (advanced)

```
<WARNING>File not yet processed: please retry later ...</WARNING>
<TRANSMISSION attributes: LEVEL, DESCRIP> Beginning of transmission
<GROUP1 attributes: ACTION, BRAND, FILE, METHOD, STATUS, DATE > Beginning of Group1, up to 3 groups are possible.
Attributes depend on the selected group.
<PAYMENT attributes: ID, REF, ORDER, STATUS, LIB, ACCEPT, NCID, NCSTER, PAYDATE, CIE, NAME, COUNTRY, TOTAL, CUR,
METHOD, BRAND, CARD, UID, STRUCT, FILEID, ACTION, TICKET, PSPID, DESC> Payment description. The attributes depend on
the selected file structure.
</TRANSMISSION> end of transmission
</DOWNLOAD_REPLY> end of file
```

Example

```
<?xml version="1.0"?>
<DOWNLOAD_REPLY>
<TRANSMISSION LEVEL="Order" DESCRIP="- Order date 12/04/2007- Status All except Cancelled by client,Invalid or incomplete-">
<PAYMENT ID="1651818" REF="order0001" ORDER="12/4/2007" STATUS="5" LIB="Authorised" ACCEPT="testoff" PAYDATE="" CIE=""
NAME="Paul Smith" COUNTRY="" TOTAL="120.00" CUR="EUR" SHIP="0.00" TAX="0.00" METHOD="CreditCard" BRAND="VISA"
CARD="XXXXXXXXXXXX1111" STRUCT=""></PAYMENT>
<PAYMENT ID="1651819" REF="order0002" ORDER="12/4/2007" STATUS="5" LIB="Authorised" ACCEPT="testoff" PAYDATE="" CIE=""
NAME="Bill Durand" COUNTRY="" TOTAL="56.00" CUR="EUR" SHIP="0.00" TAX="0.00" METHOD="CreditCard" BRAND="Eurocard"
CARD="XXXXXXXXXXXX9999" STRUCT=""></PAYMENT>
</TRANSMISSION>
</DOWNLOAD_REPLY>
```

Fixed length

Layout

```
<DOWNLOAD_REPLY> Beginning of the file
<TRANSMISSION Description string>
Sequential list of payments: each line corresponds to one order. The length of each field is described one page you can check via the
"More info" link on the Electronic reporting page.
<END_TRANSMISSION> End of the payments selection
```

Example

```
<DOWNLOAD_REPLY>
<TRANSMISSION Order Level: - Order date 12/04/2007- Status All except Cancelled by client,Invalid or incomplete->
1651818 order0001 12/4/2007 5 Authorised testoff Paul Smith 120.00EUR 0.00 0.00 CreditCard VISA XXXXXXXXXXXX1111
1651819 order0002 12/4/2007 5 Authorised testoff Bill Durand 56.00EUR 0.00 0.00 CreditCard Eurocard XXXXXXXXXXXX9999
<END_TRANSMISSION>
```

5.2.3 Extra features

Grouping payments together

You can group payments together in your download, according to certain criteria/values. You can set these criteria in the drop down lists

Upload/download files with Ingenico ePayments Batch (advanced)

under "Sort by" in the "Electronic Reporting page". Three different values can be selected, allowing the merchant to create sub-groups. Within a group, payments will be sorted according to the value of the sub-group. For instance:

```
<GROUP1 FILE="1">
  <GROUP2 STATUS="5">
    <GROUP3 BRAND="EUROCARD">
      <END_GROUP3>
    <GROUP3 BRAND="VISA">
      <END_GROUP3>
    <END_GROUP2>
  <GROUP2 STATUS="2">
    <GROUP3 BRAND="EUROCARD">
      <END_GROUP3>
    <GROUP3 BRAND="VISA">
      <END_GROUP3>
    <END_GROUP2>
  <END_GROUP1>
```

The following example illustrates a Standard (structure), Delimited (format), Sort by File (drop down 1) and Brand (drop down 2) download.

Example

```
<DOWNLOAD_REPLY>
  <TRANSMISSION Order Level: - Order date 12/04/2007- Status All except Cancelled by client,Invalid or incomplete->
  <GROUP1 FILE="40555">
    <GROUP2 BRAND="EUROCARD">
      1651819;order0002;12/4/2007;5;Authorised;testoff;;;Bill
      Durand;;56.00;EUR;0.00;0.00;CreditCard;Eurocard;XXXXXXXXXXXX9999;;
      1651821;order0004;12/4/2007;5;Authorised;testoff;;;Mark van
      Steen;;33.00;EUR;0.00;0.00;CreditCard;Eurocard;XXXXXXXXXXXX9999;;
    <END_GROUP2>
    <GROUP2 BRAND="VISA">
      1651818;order0001;12/4/2007;5;Authorised;testoff;;;Paul Smith;;120.00;EUR;0.00;0.00;CreditCard;VISA;XXXXXXXXXXXX1111;;
      1651820;order0003;12/4/2007;5;Authorised;testoff;;;Jef
      Geeraerts;;75.10;EUR;0.00;0.00;CreditCard;VISA;XXXXXXXXXXXX1111;;
    <END_GROUP2>
  <END_GROUP1>
  <GROUP1 FILE="40558">
    <GROUP2 BRAND="EUROCARD">
      1651834;order5553;12/4/2007;5;Authorised;testoff;;;John
      Smith;;75.10;EUR;0.00;0.00;CreditCard;Eurocard;XXXXXXXXXXXX9999;;
    <END_GROUP2>
    <GROUP2 BRAND="VISA">
      1651835;order5555;12/4/2007;5;Authorised;testoff;;;John Doe;;50.32;EUR;0.00;0.00;CreditCard;VISA;XXXXXXXXXXXX1111;;
      1651836;order5558;12/4/2007;5;Authorised;testoff;;;Jane Doe;;86.42;EUR;0.00;0.00;CreditCard;VISA;XXXXXXXXXXXX1111;;
    <END_GROUP2>
  <END_GROUP1>
<END_TRANSMISSION>
```

Headers

Upload/download files with Ingenico ePayments Batch (advanced)

You can indicate whether you want to have payment records with or without headers.

A file without headers contains only the transaction lines. If you want to use the <DOWNLOAD_REPLY>, <TRANSMISSION> or <GROUP> tags, you need to enable the "Headers" box.

6. Automatic upload and download test

Our server hosts several test pages (URLs) which you can call up in your browser to simulate the upload process. These URLs are for test purposes only, for establishing the data your system has to submit and the data you receive in response:

Up/download	Action	Test URL in the browser
Upload	Send and check	https://ogone.test.v-psp.com/ncol/test/AFU_step1.asp
Upload	Process	https://ogone.test.v-psp.com/ncol/test/AFU_step2.asp
Download	Check file status	https://ogone.test.v-psp.com/ncol/test/AFU_step3.asp
Download	Download transaction file	https://ogone.test.v-psp.com/ncol/test/AFU_step4.asp

7. File statuses

Default file statuses:

- Uploaded: the file has been received, but has not yet been validated (SEND step).
- To be checked: the file is awaiting validation. (when you request PROCESS in ASYNC mode for an "Uploaded" file).
- Checked: the file has been validated (CHECK step).
- Cancelled: the file has been cancelled after the 'check' step.
- To be loaded: the file is waiting to be loaded into our process module (when you request a PROCESS in ASYNC mode for a "Checked" file).
- Being loaded: the file is being loaded into our process module.
- Loaded: the file has been loaded into our process module, but (all) the payments have not yet been sent to the acquirers/banks.
- Processed: all the payments in the file have been sent to the acquirers/banks.

Optional (the "implicit batch split" option has to be enabled for your account), for files that contain more than 2750 lines:

- To be split: the file meets the generic criteria and will be sent for splitting.
- Being split: the file is currently being split.

8. Payment method specifics

For certain payment methods, the BRAND/CARDNO/ED values differ from the standard credit card values.

8.1 Direct Debits

8.1.1 Direct Debits AT

The following are the specific BRAND/CARDNO/ED field values that enable Direct Debits AT transactions to be transmitted via Batch.

Field no.	Field	Specific value
3	BRAND	"Direct Debits AT"
4	CARDNO	Bank account number. Format: XXXXXXXXXXXBLZYYYYY XXXXXXXXXX: account number, numeric, 11 digits. YYYYY: Bank code (Bankleitzahl), 5 digits.
5	ED	"99/99" or "9999"

8.1.2 Direct Debits DE (ELV)

The following are the field values that enable ELV transactions to be transmitted via Batch:

Field no.	Field	Description	Mandatory
3	BRAND	"Direct Debits DE"	Y
4	CARDNO	Bank account number + BLZ. Format: XXXXXXXXXBLZYYYYYYYY XXXXXXXXXX: account number, numeric, 1 to 10 digits. YYYYYYYYY: Bank code (Bankleitzahl), 8 digits. OR IBAN account number: 22 alphanumeric chars (SEPA)	Y
5	ED	"99/99" or "9999"	Y
Optional for SEPA (*) transactions (not applicable for Billpay and Wirecard):			
39	MANDATEID	Unique mandate reference. Telego: Format: max. 35 AN characters (charset: "A-Z a-z 0-9 space /-?:(),'+") If not provided, the platform will take the ORDERID or PAYID Easycash: Format: max. 27 AN characters (charset: "A-Z a-z 0-9 space /-?:(),'+") If not provided, easycash will generate a value.	N

(*SEPA: Single Euro Payments Area)

8.1.3 Direct Debits NL

The following are the field values that enable Direct Debits NL transactions to be transmitted via Batch.

Field	Field	Description	Mandatory
-------	-------	-------------	-----------

Upload/download files with Ingenico ePayments Batch (advanced)

no.			
3	BRAND	"Direct Debits NL"	Y
4	CARDNO	IBAN account number: max. 35 alphanumeric characters OR Regular Dutch account number: max. 10 characters (if less, left pad with zeros).	Y
5	ED	"99/99" or "9999"	Y
Only relevant for SEPA (*) transactions:			
8	CN	Bank account holder's name	Y
10	OPERATION	Action to be performed. Possible Values: <ul style="list-style-type: none"> SAL/SAS: debit money from the bank account RFD/RFS: credit money to the bank account / refund 	Y
38	BIC	Bank Identifier Code. Format: max. 11 AN characters	Y
39	MANDATEID	Unique mandate reference. Telego: Format: max. 35 AN characters (charset: "A-Z a-z 0-9 space /-?:().,.'+'") If not provided, the platform will take the ORDERID or PAYID Easycash: Format: max. 27 AN characters (charset: "A-Z a-z 0-9 space /-?:().,.'+'") If not provided, easycash will generate a value.	Y
40	SIGNDATE	Date mandate was signed by the buyer. Format: YYYYMMDD If not provided, the transaction date will be taken	Y
41	SEQUENCETYPE	Possible values to indicate the Direct Debit transaction type (max. 4 AN digits): <ul style="list-style-type: none"> "FRST": First collection of a series of Direct Debit instructions "RCUR": Direct Debit instructions where the debtor's authorisation is used for regular Direct Debit transactions initiated by the creditor "FNAL": First collection of a series of Direct Debit instructions (afterwards same MandateID can't be used anymore) "OOFF": Direct Debit instruction where the debtor's authorisation is used to initiate one single Direct Debit transaction If not provided, the transaction date will be taken	Y

(*SEPA: Single Euro Payments Area)

8.2 Payment methods with only maintenance via Batch

For certain (non-credit card) payment methods, you cannot send new transactions via Batch, but you can send certain maintenance operations via Batch.

This is the case for PostFinance Card, PostFinance e-finance, PayPal Express Checkout and TUNZ.

When sending maintenance operations, BRAND/CARDNO/ED are not required data, so no specific values need to be sent for these payment methods.

8.3 Maestro/MasterCard Advanced Registration Program

Activation

You need to contact our Sales team or your dedicated account manager to check if you are eligible to use the MARP feature. One of the conditions that need to be met is that your acquirer supports MARP.

Maestro/MasterCard Advanced Registration Program (MARP) is a system that enables you to process subsequent transactions of returning customers, without each time having to perform a 3-D Secure check.

This feature is particularly interesting for merchants who have many returning customers or who want to do regular invoicing with permission of the customer.

The eventual risk is on the expense of the merchant, as he is responsible for monitoring the first 3-DS or PIN confirmed transaction.

As a 3-D Secure check is required for the initial transaction (via e-Commerce or DirectLink), only subsequent transactions can be processed via Batch.

Position 37 of the transaction line is foreseen for the "ARP_subsequent" field, in which you need to enter the value "1" (any other value will be considered as "0" and as such of no value).

The ECI value (position 35) can only be 9 (=Recurring after first e-commerce transaction).

Example

```
1213;EUR;;5399999999999999;12/25;Order0001;;Paul Smith;;SAL;;;;;;;;;;;;123;;;;;;;;;9;;1;; (9: ECI, 1: ARP_subsequent)
```

In the Ingenico ePayments back office, when you check the order overview of a transaction, either the "ARP_initial" or the "ARP_subsequent" value (in case of Batch) will be displayed.

9. Other file formats

9.1 Purchasing cards (e-Supply)

The structure and format of e-Supply payment files should comply with a few basic rules:

- The file should be a Text ASCII file
- Several lines per invoice/order (except for transactions without Line Item Details) with the lines separated by the Carriage Return and Line Feed characters (ASCII : 13 10 – HEX : 0xD 0xA)
- Each line must contain relevant fields (corresponding to the record TYPE) separated by a semicolon (“;”)
- The fields themselves cannot contain any semicolon (“;”)
- File structure:

The file can include several invoices/orders. For each invoice/order, you have to use a group of several records:

- INV: Invoice/Order Header record (1 record per invoice/order)
- CLI: Client details record (Optional, 0 or 1 record per invoice/order)
- DET : Invoice/Order Line Item details records (Mandatory, 1 to x per invoice/order)

```
Header
INV ;..... ;
DET; .....;
DET; .....;
DET; .....;
INV ;..... ;
CLI ;.....;
DET; .....;
DET; .....;
INV ;..... ;
DET; .....;
DET; .....;
... ..
Footer
```

Please refer to the e-Supply Excel spreadsheets (available via Support > Integration & user manuals in your account) for a detailed description of the INV, CLI, and DET records.

9.2 Others

We can accept other file formats such as CARUS, DMP and PLINK. Please refer to the corresponding format specifications for these file formats.

To indicate you are using a file format that differs from the default format described in this document, you can use the following header:

```
OFM;FILE_FORMAT;
```

For instance:

```
OFM;NCSERVER;
```

If you send your files automatically, this information can only be sent as a header, not as a parameter.

Upload/download files with Ingenico ePayments Batch (advanced)