



IMACASH³

Manual: How to Integrate

Contents

What is ImaCash?	3
Reading the database	4
Writing in our database	5
Files	6
TCP/IP Socket	7
ERROR CODE TABLE	8
WORKFLOW TCP/IP	9
WORKFLOW FILES	10

What is ImaCash?

ImaCash is a software that acts between the cash drawer and the point of sale software, so that we consult the data at the origin of the sales, analyze them, show their sale to make payment and store their result.

It should be emphasized that **ImaCash** is not only an integration solution between the different cash control machines and the sales control of the establishment, **ImaCash** aims through various mechanisms, to stop fraud in the establishments.

ImaCash can be integrated by different methods. Each one of them is completely valid for a pure integration, but it is necessary to bear in mind that, depending on the method used, we will lose robustness, being the **reading to the database** the best of all.

Our software collects the sales made every 3 seconds (configurable), comparing the differences and collecting only the sales that have not been collected previously.

We have several methods to carry out integrations:

- Reading the database
- Writing in our database
- Files
- TCP/IP Socket
- OCR (*Optical Character Recognition*)

Reading the database

The client must create a user of the *POS database* that only has reading permission and that shows us which are the sales tables, vendors, counters and companies. With that user, we'll do the integration. We also give the option for the client to create views of the *POS database* with the tables mentioned above.

The *POS developer* must specify which type of database engine is being used (*MySQL, SQL SERVER, Access...*), as well as the connection protocol (*TCP/IP, ODBC...*) and any non-standard parameters needed for connection.

The data we need to read from the database is:

- Ticket number
- Date and time
- Total
- User
- Counter
- Cash or card
- Company or activity (not necessarily strict).

This is the method we recommend the most, as it allows us to read the database, we can guarantee the complete functioning of ImaCash, making integration fast, simple and safe.

Writing in our database

In our database engine, the *POS* can write or copy all *POS* sales transactions. This option is valid if the developer does not want to provide data from his connection or the database engine he uses is not standard and prefers to write this data in our *SGBD* engine.

In this case you can configure our database for full access and create a database in our *MySQL* similar to this one:

SALES TABLE

- Id varchar (30) PK
- Timestamp date - with precision up to seconds
- Company integer (11)
- Seller integer (11)
- Integer counter (11)
- Amount integer (11) - what is charged in cash (preferably in cents)
- Type integer (1) - 0 if it is in card, 1 if it is in cash

SALES TABLE

- Id integer (11) PK
- Name varchar (250)

TABLE OF COUNTERS

- Id integer (11) PK
- Name varchar (250)

COMPANY TABLE

- Id integer (11) PK
- Name varchar (250)

Files

The client is the one who writes to us and sends us files of any format (*txt, xml, json...*) using optionally *AES 256* encryption.

For this method, it is necessary to have a **shared folder on the network**, which has full **ImaCash** and *POS* permissions for modification, creation and deletion.

If the files are *txt*, there will be a single line indicating the operation in question with the following format (separated by ";").

(Amount);(POS/CASH number);(Invoice/ticket number);(Number/User name);(Date and time);((1 if paid in cash)/(0 if paid by card)).

Depending on the type of transaction, the file name will change.

Example

Sale made by Carlos of 1€, at the MOST2 counter at 12:12:25 on 1 January 2017. The ticket number is 12345 and has been paid in cash:

- **V12345.txt >> 1,00;MOST2;V12345;CARLOS;2017-01-01 12:12:25;1**

Return (or subscription) by Javier of -1€, at the MOST1 counter at 15:25:36 on 1 January 2017. The ticket number is 12346 and has been made with a card:

- **V12346.txt >> -1,00;MOST1;V12346;JAVIER;2017-01-01 15:25:36;0**

When **ImaCash** starts, it will read the pending files and load them. When the sales have been paid for, it will delete the files and create another one with the transaction response (whether it was successful or not).

Example

Successful transaction:

- **V12345.txt >> OK**

Unsuccessful transaction:

- **V12345.txt >> KO**

TCP/IP Socket

This method of integration is the least recommended.

ImaCash will work as a server and will receive payments from the different *POS* in the establishment. If **ImaCash** is switched off, the transaction made will not be recorded anywhere and will therefore be lost.

In any case, in order to carry out this type of integration, the client will send us a Socket with a single line the same as that of the *txt* files.

To send a transaction to **ImaCash** this string will be sent:

NEW TRANSACTION:

- DATA;{ID_TRANSACTION};{SALESMAN};{POS};{AMOUNT};{TYPE}
 - ID_TRANSACTION
 - It is the identifier of the transaction
 - SALESMAN
 - It's the name of the salesman
 - POS
 - It is the name of the Point of Sale (Terminal)
 - AMOUNT
 - Is the amount of transaction (integers)
 - TYPE
 - It's the type of transaction
 - 0 = Card
 - 1 = Cash
 - Examples:
 - DATA;V000001;CARLOS;P1;120;1
 - Transaction V000001 created by Carlos in POS P1, amount 1,20€ in cash
 - DATA;V000002;JAVIER;P2;562;0
 - Transaction V000002 created by Javier in POS P2, amount 5,62 € in card

CANCEL: (POS-side)

- DATA;{ID_TRANSACTION};CANCEL

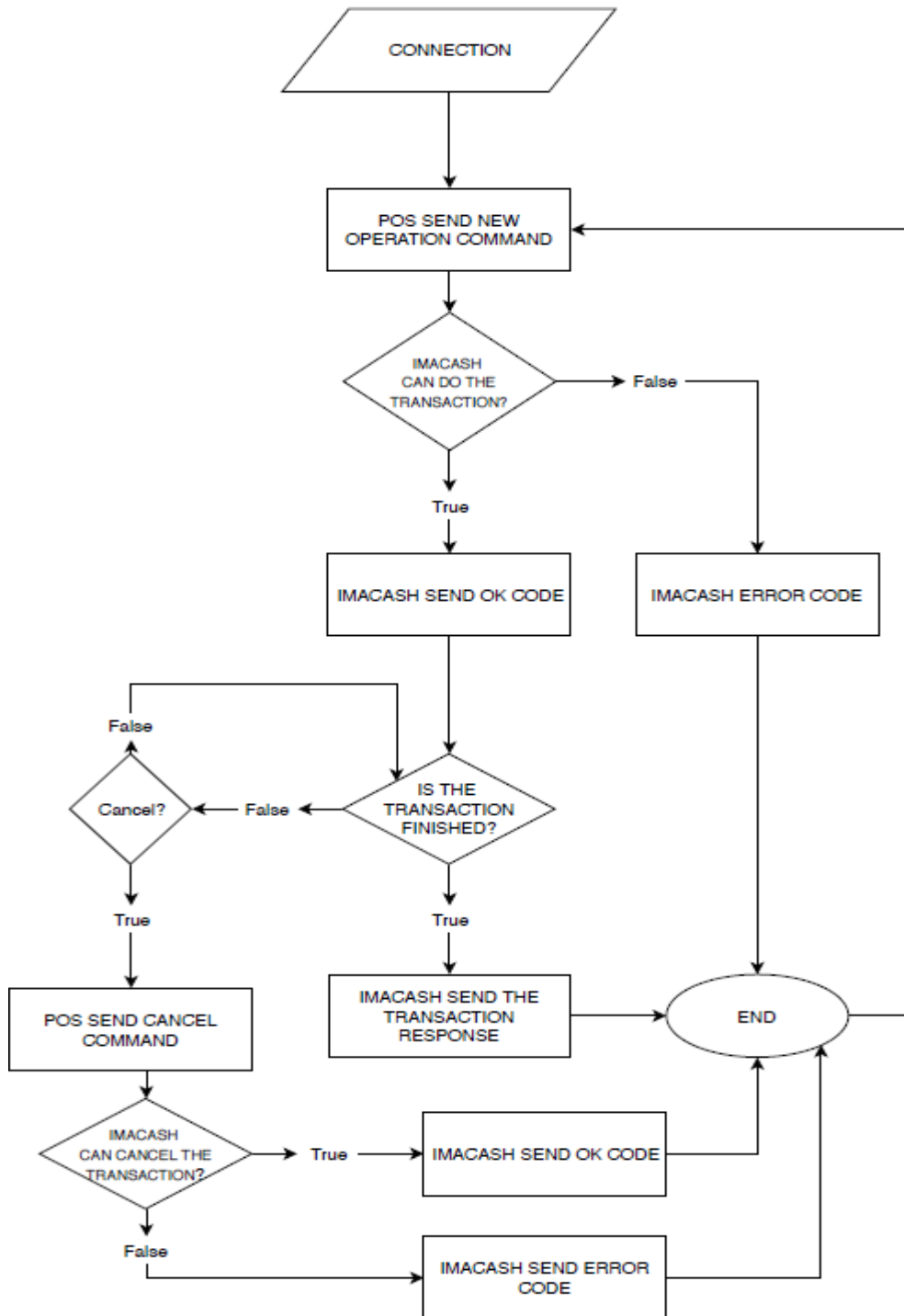
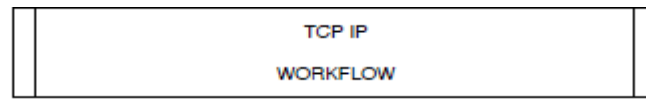
RESPONSE: (ImaCash-side)

- DATA;{ID_TRANSACTION};{CODE}
 - ID_TRANSACTION
 - It's the identifier of the transaction.
 - CODE
 - The response code (see code table);
 - If the code is **000**, the transaction is fine otherwise, there is an error.

ERROR CODE TABLE

NAME OF THE FIELD	CODE	DESCRIPTION
SUCCESS	000	The operation was successful.
PAID	001	The operation is paid for.
UNKNOWN_ERROR	999	The operation failed for a reason not specified above. The error message field can be checked for more information.
ERROR_TICKET_ID	101	The number of entries is empty or has special characters.
ERROR_TICKET_TYPE	104	The ticket type is different from 0 or 1.
ERROR_TICKET_CASHIER	105	The cashier's name is empty or has special characters.
ERROR_TICKET_POS	106	The name of the device cashier is empty or has special characters.
ERROR_TICKET_AMOUNT	107	The total quantity is empty or is not a number.
ERROR_TICKET_DUPLICATED	108	This operation is already in the database of the executing or completed BP.
ERROR_TICKET_NOT_EXISTS	109	The Requested Ticket does not exist

WORKFLOW TCP/IP



WORKFLOW FILES

