Virtual Plus POS

Operating and Installation Manual

Sabadell

Index

1.
Introduction

Banco Sabadell is the bank chosen by the best in merchant and as such, is the leader in merchant collection solutions, always anticipating and conducting ongoing research into the most advanced technological media.

eCommerce is no longer exclusive to a certain type of company: small merchants, professionals, SMEs, major corporates, etc., an increasingly large number of companies are adopting eCommerce and require **secure solutions which adapt** to thee reality of their merchant.

In our opinion, this requires a technology capable of delivering multiple requirements. In short, virtual POSs which meet the needs of any company or merchant operating online.

For this reason, **Banco Sabadell has strengthened its eCommerce service** and has a specific unit of agents specialised in virtual payment gateways and a back-office team to offer our customers different secure solutions together with a broad range of services in online sales.

Two types of needs, two POS solutions

Banco Sabadell offers two payment gateways in line with the characteristics of the customer:

- **Virtual POS**. The most widely used solution which effectively covers the requirements of merchants and SMEs. This gateway is easy to install yet offers a broad range of services and specific benefits for eCommerce.
- **Virtual Plus POS**. This is a more sophisticated solution which is designed for companies with a high volume of online sales. It offers an advanced set of technical and operating services in addition to ongoing support by agents specialised in eCommerce payments. **This manual covers the descriptions and installation instructions of the Virtual Plus POS solution services**.

In addition, Banco Sabadell has an additional Virtual POS solution called **Virtual POS Institutions**. It is a payment gateway designed specifically to meet the needs of organizations and public institutions wishing to offer the payment service notifications and payment of taxes, directly from their website.

Open Source Tools

We make available, free of charge, a selection of the best OpenSource tools available for the area of e-commerce.

With them, you can set yourself up your online store and easily manage their look and feel, usability and functionality, in addition to integrating with Virtual POS Banco Sabadell much easier.

Ask our technical support to get more information or the integration manuals (see section 7.11 of the manual)

Prestashop, Magento, Wordpress Woocommerce, OsCommerce, Zencart, Opencart, Wordpress E-Commerce and Virtuemart (Joomla).

Security elements

Maximum security is one of Banco Sabadell's top priorities. Our integral **3D Secure gateway** (Secure Electronic Purchase) which operates under international protocols verified by Visa and MasterCard SecureCode (both based on Secure 3D technology), offers high security and payment protection.

These protocols obtain the **authentication of the holder** when making the purchase, i.e. the customer is identified as the legitimate holder of the card being used.

However, there are establishments which prefer to deactivate the 3D Secure protocols and replace them with alternative **fraud control** systems. In this case, they can simply request their bank agent to analyse the merchant and implement the modification if they consider it appropriate.

Likewise, and especially for merchants and SMEs, the Virtual POS of Banco Sabadell is configured with security limits – velocity checks – which validate repeated attempts to purchase with the same card and/or from the same IP, significantly reducing the risk of fraud.

The security requirements are even more rigorous in the case of the Virtual Plus POS, in accordance with the high billing volumes. Specifically, it includes additional security elements such as: **Advanced fraud management rules, daily reports** on doubtful transactions (claimed, disputed or declared illicit by the purchasers) and **collaboration and technical integration** agreements with major gateways, processers and international fraud-scrubbing companies.

Payments of subscriptions and Express Payments: Enhancing the user experience

The Banco Sabadell Virtual POSs accept the usual operations: Authorisations, pre-authorisations, authentications, returns management and recurring purchases.

But the true innovation lines in the system via which the card details are stored on the gateway itself.

The advantage is clear: with this functionality the merchant customer enters his card details just once on making the first purchase and does not need to repeat this step in future payments in the same establishment. Thus, the merchant increases its website usability (express payment) and also has a tool for processing subscriptions or other regular payments.

Solutions for internationalisation

In eCommerce the limit is the world. Banco Sabadell has taken special care of this feature, integrating solutions which facilitate cross-border sales:

- The **Multicurrency service** allows customers to purchase in a **wide range of local currencies**, avoiding the obstacles usually associated with currency conversion.
- **DCC operations** (Dynamic Currency Conversion) offers **online conversion of the local currency into the Euro**. This operation is launched as soon as the Virtual POS detects that the card has been issued outside the Eurozone.
- The gateway is also **multi-language**, both for the merchant and the purchaser. Currently the Virtual POS accepts operations in Spanish, Catalan, Basque, English, French, German, Portuguese, Dutch, Polish, Italian and Swedish.

There are also specific tools for the Virtual Plus POS which have been developed to maximise sales and simplify transactions via international affiliates:

- Many countries have l**ocal payment systems which are different** to financial cards, which are widely accepted. These sales cannot be lost and for this reason, Banco Sabadell has international agreements to access a large number of these payment systems.
- If the company has affiliates in other European countries, **thanks to the Banco Sabadell Cross-Border license** it is possible to process Visa or MasterCard pay-

ments at Spanish merchants and affiliates. A single integration with the Virtual Plus POS enables all sales to be managed.

Back Office Tools

We believe the merchant management should be simply and user-friendly, but also a complete solution. The Virtual POS includes a website-based administration module designed to offer simple use and offer all the functionalities.

- **Real time control** of all operations.
- **Access to the account closes**, with permanent availability of those for the last year.
- Maximum **simplicity** in returns management.
- **List of transactions**, which can be downloaded to the computer and which includes all the important information.
- For large companies, **integration with the corporate intranet or proprietary applications** and availability of files Via FTP and BS Online.

2.
What do I need?

2.1 How is it installed?

The first step in installing the Banco Sabadell Virtual POS is to process apply for a merchant contract and registration of the Virtual POS at your branch.

To contract this service you need to provide us with some basic details of the merchant and your virtual store.

Once the application has been accepted, you will be sent an email with the unique security codes for your merchant to enable you to install the Virtual POS. In order to expedite the integration of the Virtual POS with your web server and synchronise the purchasing mechanisms, before implementing the Virtual POS in real time, we recommend using the codes in a test environment included in this manual.

In the event of any doubt or enquiry, the Banco Sabadell Virtual POS Telephone Support Service will be available to help you via email or over the telephone.

2.2 What should my website have?

In accordance with the requisites of the card brands (Visa and MasterCard) and the Bank of Spain, any online store with a virtual PoS must have:

1. Shopping cart or similar where buyers request the purchase of the product or service. For this purpose, Banco Sabadell will require the website is accessible and allows a trial purchase (in the case of webpages under construction, the merchant must provide access to its test environment).

2. The "Legal Notice" (or similar section) must contain the business name, identification (CIF), registered office and contract details of the merchant.

3. The "Terms and Conditions" (or similar section) must contain the policy on returns.

4. Should the merchant work with a payment services provider, the latter must first be authorised by Banco Sabadell.

5. The domicile and country of establishment must appear on one of the pages the holder accesses during the payment process (it must be visible and never linked to an external website).

The above requisites will be verified by the Banco Sabadell team during the virtual PoS registration process. If it is not implemented, we shall contact to merchant for its modification. To avoid delays, we ask the merchant to verify it meets the above requisites during the registration process.

When the purchaser chooses to pay using a credit card, the Banco Sabadell Virtual POS is activated.

2.3 What payment regulations must I follow?

The Virtual POS, given its nature, is subject to rules arising from its participation in international payment systems and its management by Banco Sabadell.

These regulations are included in the contract signed by Banco Sabadell and the merchant. Please take note of the following rules:

- The merchant may only process transactions originating from the Webpages duly verified by Banco Sabadell.

- The merchant shall immediately cancel card operations when an undue charge occurs or when the sales process and goods delivery has not been completed.

- The merchant will not, under any circumstance, store the card details onsite, except those necessary for operation, in which case it will be subject to the Security program PCI/DSS of VISA and MASTERCARD. Even in this case, it is strictly forbidden to store the CVV2 code (three security digits stamped on the back of the cards) under any circumstance.

3.
Security measures

The Virtual POS associated with your merchant has been configured with a series of security measures to reduce the risk of sales paid for with fraudulent cards (stolen, copied or used without the authorisation of the legitimate holder).

3.1 Velocity checks

These are security restrictions which block unusual operations or purchasing behaviour.

As an additional security and fraud prevention measure, Banco Sabadell will apply a series of security limits on the merchant's operations in keeping with its activity and types of operation. They are limits on the amount and number of operations, which must conform to certain values which do not affect the sales expectations of the merchant but avoid exaggerated deviations from the usual turnover (in most cases they mean that an attack is in progress using stolen and/or fraudulent cards).

There are limits on the basis of the following parameters:

- Maximum number of operations (accepted and denied) per card
- Maximum number of operations (accepted and denied) per user (IP address)
- Maximum amount accumulated per card
- Maximum amount accumulated per user (IP address)

If you consider that these parameters do not conform to your usual merchant operations, please request a modification via your branch or your Banco Sabadell agent

In addition, other rules can be configured in line with the amounts, number of operations, country where the card was issued, country where the IP location of the purchaser, period of use, etc.

If you consider that these parameters do not conform to your usual merchant operations, please request a modification via your branch or your Banco Sabadell agent.

3.2 Verification of the CVV2

The CVV2 is a three-figure code stamped on the back of all financial cards. Validation of this code has proven to be an excellent tool for fraud limitation.

The Banco Sabadell Virtual POS will always request the CVV2 code during the payment process and will validate it online with the financial entity which issued the card.

3.3 3D Secure Protocol

In order to protect the merchant from fraudulent payments or chargebacks of purchasers arguing that they did not make the purchases, all the virtual POSs of Banco Sabadell are certified under the Secure Electronic Commerce protocols (3D Secure) of the Visa (Verified by Visa) and Mastercard (MasterCardSecureCode) card systems.

In 3D Secure, within the payment process, Banco Sabadell requires the cardholder to authenticate themselves online with their financial entity. The authentication system is first agreed on with the holder and their bank (password, PIN, verification SMS, etc.)

To be taken into account:

- Although the 3D Secure offers security and protection, **if a virtual merchant has alternative fraud control systems and wishes to deactivate the 3D Secure purchase of their Virtual POS, they can request this from their branch or Banco Sabadell** agent to analyse the case and implement the modification if appropriate.

- Card systems do not usually allow company cards (Merchant, Corporate, etc.) to carry out the holder authentication process. For this reason, this type of card is not accepted by the virtual POS. In the exceptional cases of a merchant considering it necessary to accept company cards, it must request this from its Banco Sabadell branch, previously and expressly accepting the charge-backs arising from these operations.
- The authentication of the cardholder does not release the merchant from accepting the charge-back of operations occurring due to other causes in which the customer argues that they carried out the transaction but, for example, claims that they did not receive the service or items paid for. To defend itself from such chargebacks, the merchant must furnish Banco Sabadell with documentation demonstrating indisputably that the cardholder received the product or service in question.

3.4 Additional security measures

To protect the interests of your merchant and reduce the number of incidents, we recommend monitoring the activity of your webpage to detect any of the following suspicious signs of fraud:

- In the Virtual POS administration module the IP address of the purchaser appears together with the card number (duly masked by asterisks). It is suspicious if:
  - _ The same user (IP address) has paid (or attempted to pay) with more than two different cards.
  - _ The same user (IP) or the same card have carried out multiple operations over a short period of time.
  - _ When making different purchases, the same user (IP) or the same card have been registered on the website with different details.
  - _ If the POS has rejected the first card operation, it is suspicious if further operations have been processed with the same IP or with the same card for lower amounts.
  - _ Consecutive operations with similar card numbers.
- In the response message ("DS_Response" field) or the Virtual POS administration module it appears if the operation has been accepted (000 to 099 codes) or rejected (other codes). 2xx type rejection codes indicate that the card is blocked due to loss, theft, forgery or fraudulent use of the card number. In these cases the merchant must block the user (identifiable via the IP address and registration details) and not allow any option of re-attempting payment.
- In the response message is the "Ds_Card_Country" field which states the ISO code of the country where the card was issued. By comparing the IP address of the buyer it is possible to filter behaviour suspicious of being fraudulent (e.g. A card issued in one country but operating via an IP of a different country).
- In the purchaser's registration information:
  - _ Check the telephone numbers by using public telephone directories.
  - _ Check whether the telephone code and/or prefix match the geographic area of the delivery address.
  - _ Check the match between the post code and city of delivery.
  - _ Check the email address by sending a confirmation order.

- _ Check in public details of social networks the purchaser's registration details.

- Also check:
  - _ Orders with the same delivery address but made with multiple cards.
  - _ Orders for multiple numbers of the same product.
  - _ Orders for an amount above usual.
  - _ Orders for which delivery is urgent or even "for the next day". Criminals wish to fraudulently obtain these products as soon as possible for probable resale and are unconcerned about the surcharge on delivery.
  - _ For websites not translated to international languages payments made with foreign cards and/or from international IPs and/or order to be sent to international addresses.

In addition to monitoring the parameters above, your merchant can significantly reduce its exposure to the risk of fraud by applying its own controls over operations to identify high risk operations. These controls can be automatic (velocity checks) and prior to sending the requests for authorisation to Banco Sabadell; or subsequent manual checks to processing the transaction with Banco Sabadell.

The anti-fraud protocols implemented must be based on the user's registration details (User ID., Name, Telephone no, Address, email, etc.) and, also on the registration details of the recipient of the service/product (name of the travellers if it is a travel agency or similar, product delivery address, contact telephone, etc.).

Should the operation not pass all the above controls, the merchant must reject the card as the means of payment and cancel the operation if already carried out via the Virtual POS.

To minimise the risk of fraud it is therefore necessary for the merchant supervisors to know these security measures, prepare training activities for all the employees handling card payments and periodically check compliance with these measures. Otherwise, there exists the risk that fraudulent operations can be charged back to the merchant and if the number of fraudulent or charged back operations is significant, the terminal is blocked and the contract with Banco Sabadell terminated.

4.
Cross border license

Banco Sabadell has cross border licenses issued by the two major card companies: VISA and Mastercard.

The cross border license allows Banco Sabadell to process the card operations performed at virtual stores of merchants and companies with tax residence abroad or Spanish multinational companies with affiliates overseas.

The advantages for merchants and companies operating in more than one European country are clear: The signing of an acquisition agreement with Banco Sabadell **will avoid the complexity and cost associated with the signing of acquisition agreements with local banks** in other European countries in which the merchant has a virtual store.

VISA: Cross border license activity area:

Merchants and companies resident in: Germany, Andorra, Austria, Belgium, Bulgaria, Cyprus, Denmark, Slovakia, Slovenia, Spain, Estonia, Finland, France, Gibraltar, Greece, Greenland, Holland, Hungary, Iceland, Faroe Islands, Ireland, Israel, Italy, Latvia, Lithuania, Luxembourg, Malta, Norway, Poland, Portugal, United Kingdom, Czech Republic, Rumania, Sweden, Switzerland and Turkey.

MASTERCARD: Cross border license activity area:

Western Europe Region

Merchants and companies resident in: Germany, Andorra, Austria, Belgium, Bulgaria, Cyprus, Vatican City, Denmark, Slovakia, Slovenia, Spain, Estonia, Finland, France, Gibraltar, Greece, Holland, Hungary, Iceland, Ireland, Isle of Man, Channel Islands, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Monaco, Norway, Poland, Portugal, Rumania, United Kingdom, Czech Republic, San Marino, Sweden, Switzerland and Turkey.

5.
Operational aspects

5.1 Types of transactions

In accordance with the needs of each merchant, the Virtual POS offers a wide range of authorisation requisitions which the merchant can combine as required.

Standard payment or authorisation
(Ds_Merchant_TransactionType = "0")

This is the most widespread case in which the transaction is initiated by the holder who is connected via the Internet to the webpage of the merchant during the payment process. Once the purchase request has been received by the merchant, the Virtual POS requests the details to perform the authorisation transaction.

If the merchant is configured as CES (Secure eCommerce) and the bank of the cardholder has an authentication system, the bank will request the cardholder proof of identification.

The request for Authorisation is carried out in real time, entailing an immediate charge to the account of the holder associated with the card (credit or debit).

Partial or Total Refund
(Ds_Merchant_TransactionType = "3")

These are book transactions initiated by the merchant, which can also use the Virtual POS administration module to perform them manually.

The Virtual POS checks for existence of the original authorisation to be refunded, and that the sum of the amounts refunded does not exceed the original authorised amount under any circumstance.

They produce a book effect on the holder's account **(some issuing entities take several days to pay the holder)** and are therefore captured automatically and sent to the settlement process of Banco Sabadell which will make the relevant charge in the merchant's account.

Pre-authorisation
(Ds_Merchant_TransactionType = "1")

In accordance with the regulations of the international cards, this operation is restricted to those merchants whose activity is one of the following: hotels, travel agencies and vehicle rental.

It can be used when at the time of the purchase the exact amount of same cannot be determined or if, for some reason, the merchant does not want the amount to be charged to the customer account immediately.

The transaction is transparent for the holder who at all times acts exactly the same as in the previous case, i.e. furnishes his details and is authenticated if necessary.

The request for pre-authorisation is carried out in real time, producing a withholding for the amount of the sale in the holder's account.

The transaction is not captured and therefore produces no book effect in the holder's account nor payment to the merchant **(in the case of debit cards certain issuing entities DO make a book entry for the holder which is automatically cancelled after several days)**.

Any Pre-authorisation must have Confirmation of Pre-authorisation within a maximum of 7 calendar days. Otherwise, it loses its validity as guarantee of payment.

To activate the Pre-authorisation service, the merchant must expressly request same from its Banco Sabadell branch.

Confirmation of Pre-authorisation
(Ds_Merchant_TransactionType = "2")

This is an inseparable supplement to the above operation.

In this transaction the holder is not connected to the merchant's website and it is always initiated by the merchant.

They must be confirmed within the maximum period established for each card brand and the amount must be less than or equal to the original amount.

This transaction is posted, automatically regularising the entry in the holder's account and sending it to the settlement process of Banco Sabadell for payment to the merchant.

Confirmation of pre-authorisation guarantees payment and conserves the conditions with regard to the secure transaction of its original Pre-authorisation.

The Virtual POS will check the existence of the original operation and the amount to be confirmed, rejecting the operation if any error exists.

Cancellation of Pre-authorisation
(Ds_Merchant_TransactionType = "9")

The holder is not connected to the merchant's website and therefore this transaction is always initiated by the merchant. It must be carried out within the maximum period established for each card brand.

The Virtual POS will check the existence of the original operation, rejecting the operation if any error exists.

Deferred Pre-authorisation
(Ds_Merchant_TransactionType = "0")

They are similar operations to pre-authorisations but are available to all sectors of merchant. Authorisation is obtained from the issuing bank in real time which requires confirming within the 72 hours following if the operation is to be definitive.

If 72 hours elapse from the day/time of the pre-authorisation without confirmation being sent, the authorisation is automatically cancelled and cannot therefore be confirmed.

Unlike traditional pre-authorisations, the amount of the Confirmation of Deferred Pre-authorisation must be exactly the same as the respective re-authorisation.

The request for pre-authorisation is carried out in real time, producing a withholding for the amount of the sale in the holder's account.

The transaction is not captured and therefore produces no book effect in the holder's account nor payment to the merchant **(in the case of debit cards certain issuing entities do make a book entry for the holder which is automatically cancelled after several days)**.

To activate the Deferred Pre-authorisation service, the merchant must expressly request same from its Banco Sabadell branch.

Confirmation of Deferred Pre-authorisation
(Ds_Merchant_TransactionType = "P")

This is an inseparable supplement to the above operation.

The holder is not connected to the merchant's website and therefore this transaction is always initiated by the merchant. It must be carried out within the 72 days following the original pre-authorisation and the amount must be THE SAME AS the original amount.

This transaction is posted, automatically regularising the entry in the holder's account and sending it to the daily settlement process of Banco Sabadell for payment to the merchant. Confirmation of pre-authorisation guarantees payment and conserves the conditions with regard to the secure transaction of its original Pre-authorisation.

The Virtual POS will check for the existence of the original operation and the amount to be confirmed, rejecting the operation if any error exists.

Cancellation of Deferred Pre-authorisation
(Ds_Merchant_TransactionType = "Q")

The holder is not connected to the merchant's website and therefore this transaction is always initiated by the merchant. It must be carried out within 72 days following the original pre-authorisation.

The Virtual POS will check the existence of the original operation, rejecting the operation if any error exists.

Authentication
(Ds_Merchant_TransactionType = "7")

This type of operation can be used by the merchant when the sale amount cannot be precisely determined at the time of the sale.

The operation is similar to the Pre-authorisation, although in this case only the first part of the operation is performed, i.e. the authentication of the holder. Request for authorisation does not occur, so the transaction is not posted and causes no withholding in the cardholder's account.

Subsequently, and within the following 45 calendar days, the merchant will send a confirmation of authentication which will complete the original operation.

Confirmation of Authentication
(Ds_Merchant_TransactionType = "8")

This is an inseparable supplement to the above operation.

The cardholder is not connected to the merchant's website and it is always initiated by the merchant.

The amount may vary from the original operation (even greater), and must be carried out within the 45 days following the original authentication.

This transaction is posted, causing an entry in the cardholder's account and sending it to the daily settlement process of Banco Sabadell for payment to the merchant.

Confirmations of authentication are stored under the same security conditions as the original authentication.

The Virtual POS will check the existence of the operation, rejecting it if any error exists.

Payment of subscriptions and Express Payments
(Ds_Merchant_Identifier)
(Ds_Merchant_Group)
(Ds_Merchant_DirectPayment)

So as to increase the conversion ratio and facilitate the purchasing process as much as possible, the Banco Sabadell Virtual PoS has an innovative function for making express payments and payment of subscriptions via an identifier equivalent to the card number.

This method facilitates purchase management for regular customers as they do not need to enter their card details in each purchase.

The buyer only has to enter the card details for the first purchase and the merchant receives, together with the payment response, an identifier to be used in subsequent purchases.

They will also be informed when the card expires and optionally the card number, duly masked, i.e. with certain digits replaced by asterisks.

The card details are stored in the Banco Sabadell servers, releasing the merchant from having to meet the PCI-DSS security requirements. (See section 11 of this manual).

Section 7.8 of this manual describes the **technical requirements for installing** this type of payment in your Virtual PoS.

Ask your branch or Banco Sabadell agent to activate the "Subscription Payments and Express Payments" service".

The Technical Support Service for the Banco Sabadell Virtual PoS will be available to resolve any questions about this type of payment. See contact details in section 7.11 of the manual.

5.2 Request for payment documentation on the part of the purchaser

In online shopping, the time when the purchase is made does not usually coincide with the time the purchaser receives the details from their bank about operations made with the credit card. If, in addition, the name of the merchant on the bank statement does not match or cannot be associated with the webpage where the purchase was made, this may cause the purchaser to doubt if it was really they who performed the transaction.

Therefore the purchaser is entitled to request the merchant the relevant documentation proving that it was they who made the purchase. The maximum period for this request is 12 months as from the operation date.

It should be taken into account that when a cardholder requests documentation, in many cases this is a prior step to sending a charge-back for the amount charged. To minimise the percentage of charge-backs received (and which may incur penalties if they exceed the ratios deemed acceptable by the control programs of the Card Brands), it is advisable for a merchant supervisor to analyse the requests for documentation and return those operations which, according to hid findings, may be fraudulent.

In these cases, **the card issuing entity may request the merchant send proof of the operation**. The request is made by sending a physical letter to the merchant with the details of the transaction. **The merchant is under the obligation to respond** within a maximum of **7 merchant days**. The response may be sent by fax to 93 368 72 91 or to the following email peticionfotocopias@bancsabadell.com

If delivery of goods takes place, the certificate of delivery of the delivery company must be attached. As a general rule **this certificate must be signed by the cardholder**, not by a third person.

As an exception, and for those cases in which it is not possible to delivery the goods to the cardholder (either due to inability to be in the place and time agreed for receipt or because it is a gift) delivery to a third person is allowed. In this case, this circumstance must be recorded on the order form which the customer completed for the merchant, with the following information:

- Authorised person, identified by name and identity document (DNI, Passport, etc.). The order must only be delivered to this person and the delivery note must include the signature of the recipient and the entry confirming that the identity document provided was checked.

- For receipt at hotels or similar; it will be necessary to identify the name and address of the hotel, and also the name and document of the guest who is to receive it. The receipt must be signed by a properly identified employee of the hotel and stamped by the latter. In addition, the proof of receipt must record verification that the recipient of the goods is staying at the hotel.

It is advisable not to specify a concrete delivery date for the goods, unless essential, but rather an interval of days, as any breach is sufficient cause for return.

In the case of a merchant offering services and not products, i.e. there is no goods de-

livery, the merchant will enter the following details on the response form

_ Name of merchant
_ Tax ID/Code of the merchant
_ Merchant Code (FUC)
_ Authorisation number
_ Operation date
_ Card number
_ Webpage address (URL)
_ Transaction amount
_ Currency
_ Name of the Purchaser
_ Description of product purchased
_ Define the policy on returns or indicate the URL where users may obtain the relevant information

5.3 International Sales

If you have the Virtual POS Plus, you should be aware that there are different payment systems for financial cards that have a high level of acceptance in some countries. In order to maximize sales, Banco Sabadell, through agreements with specialized international companies, offers access to a large number of local payments.

In addition, Banco Sabadell also features a cross border license issued by Visa and MasterCard to process transactions done in both Spanish and European merchants. Thus, a multinational company can manage sales of all its subsidiaries in other countries with a single integration to POS Virtual Plus.

If you want your products can be purchased from almost anywhere in the world, the Virtual POS Banco Sabadell also offers several advantages in this regard:

Multilanguage

Additionally, for better universal payment process, all pages of the Virtual POS and messages to the cardholder are available in the following languages: Spanish, Catalan, Valencian, Galician, Basque, English, French, German, Portuguese, Dutch, Polish, Italian and Swedish. The Ds_Merchant_ConsumerLanguage field (see Annex I of the manual), allows you to select the language of the payment page of the Virtual POS.

Multicurrency Service

You can target your customers by offering them the opportunity to buy in their own local currency. The multicurrency service is available to any currency in the world. Through field Ds_Merchant_Currency can choose the currency in which the payment is processed. You can view the list of currencies in Annex V of this manual.

DCC Operations (Dynamic Currency Conversion)

With this option, when the Virtual POS detects that the card being used was issued in a country other than the euro, automatically displays an information screen in which the cardholder can choose whether to pay in Euros or through an online conversion amount, in any other currency of his choice. In both cases, funds will be credited in merchant's account in euro.

Multicurrency settlement

Sell your products anywhere in the world and receive funds into your account in the original currency of the payment. You can open accounts in any of the currencies accepted in official Exchange markets, without having to worry about the exchange rate differences.

Banco Sabadell will credit your account in the same currency as the original sale transaction.

5.4 DCC Operations (DYNAMIC CURRENCY CONVERSION)

The Virtual POS of the Bank enables the holders of Visa or MasterCard cards issued in a currency other than the euro to pay for the purchases in the same currency as the card. This is a type of collection that the merchant can offer its customers.

Characteristics

This operation only applies to Visa or MasterCard cards issued in a currency other than the euro.

It is very simple to operate: When the Virtual POS automatically detects that the card can operate in DCC mode, it displays the following screen:

(*) Example based on an operation carried out with a card associated with USD currency

The screen displays the following information:

- Amount of the operation in euros. If the customer chooses this type, the transaction is performed in euros.
- The amount of the operation in the currency of the card. In this case the transaction is performed in the original currency of the card. The screen includes the mark up for a multicurrency transaction to be applied and the definitive amount of the operation, as neither the bank nor Visa or MasterCard can apply any other type of charge.

With the screen visible, the customer must freely choose whether to make payment in euros or in the currency of the card. The Virtual PoS shows the receipt for the transaction on the screen below.

Regardless of the customer's choice, the settlement to the merchant will be in euros. In the settlement statement of the merchant the following will appear: the nominal amount of the operation and the difference between the discount charge applied to the merchant for operating with the POS and the agreed bonus with the merchant for operating in DCC mode.

Advantages for the cardholder:

- The cardholder, at the time of making the purchase, knows the exact amount the bank in his country will debit for the operation.
- The cardholder's bank cannot apply any charge for currency exchange and the transaction is carried out in the original currency of the card.

Advantages for the establishment:

- Possibility of offering a service to its customer so they know the amount of the operation in advance.
- It obtains a bonus on the discount charge for operating with the POS in multicurrency operations.
- Unlike the Multicurrency mode, in which prices must be updated on a daily basis, in DCC mode the conversion is performed automatically.

As an example, section 7.9 of this manual includes a description of the access methods and the messages for payment request, confirmation and DCC response.

In the event of any doubt or enquiry, the Banco Sabadell Virtual POS Technical Support Service will be available to help you via email or over the telephone. See contact details in section 7.11 of the manual.

5.5. Flexible Descriptor

This functionality enables the merchant to add information about the operation in progress so as to help the cardholder to identify the purchase in the statement of card operations while also avoiding any possible chargebacks to the merchant.

To activate this functionality, the merchant must contact the Technical Support Service for the Banco Sabadell Virtual PoS (section 7.9).

5.6. 0 amount operations

0 amount operations allow the merchant to verify the authenticity of a card with the issuer without effecting a charge. In addition, the merchant can request a reference be generated for the card during verification.

Use of 0 amount operations

To use the 0 amount operation the merchant has three options for effecting the request:

1. Build the request by entering the parameters of the card details:

– Ds_Merchant_Amount = 0

– Ds_Merchant_Pan

– Ds_Merchant_ExpiryDate

– Dsmerchantcvv2 (Optional in accordance with the merchant's configuration)

2. Make a request by entering the parameters of the card details and generation of a reference:

– Ds_Merchant_Amount = 0

– Ds_Merchant_Pan

– Ds_Merchant_ExpiryDate

– Ds_Merchant_Cvv2 (Optional in accordance with the merchant's configuration)

– Ds_Merchant_Identifier = REQUIRED

3. Make the request by entering the reference generation parameter:

– Ds_Merchant_Amount = 0

– Ds_Merchant_Identifier = REQUIRED

This last option signifies the customer must be present during the operation as they have to enter the card details on the screen.

Example of request and WS response

Below is an example of requesting a reference be generated for the card during verification.

Request

```
<REQUEST>
<DATOSENTRADA>
<DS_MERCHANT_MERCHANTCODE>999008881</DS_
MERCHANT_MERCHANTCODE>
<DS_MERCHANT_TERMINAL>871</DS_MERCHANT_
TERMINAL>
<DS_MERCHANT_AMOUNT>0</DS_MERCHANT_AMOUNT>
<DS_MERCHANT_ORDER>1467310037</DS_MERCHANT_
ORDER>
<DS_MERCHANT_CURRENCY>978</DS_MERCHANT_
CURRENCY>
<DS_MERCHANT_PAN>491671******0017</DS_
MERCHANT_PAN>
<DS_MERCHANT_EXPIRYDATE>****</DS_MERCHANT_
EXPIRYDATE>
<DS_MERCHANT_CVV2>***</DS_MERCHANT_CVV2>
<DS_MERCHANT_TRANSACTIONTYPE>0</DS_MERCHANT_
TRANSACTIONTYPE>
<DS_MERCHANT_IDENTIFIER>REQUIRED</DS_MERCHANT
IDENTIFIER>
</DATOSENTRADA>
<DS_SIGNATUREVERSION>HMAC_SHA256_V1</DS_
SIGNATUREVERSION>
<DS_SIGNATURE>On95/3kZI9xI3/dz3/
hO8yktiFxZRWK3mOlGcmR8+qA=</DS_SIGNATURE>
</REQUEST>
```

Response

```
<RETORNOXML>
<CODIGO>0</CODIGO>
```

<OPERACION
<Ds_Amount>0</Ds_Amount>
<Ds_Currency>978</Ds_Currency>
<Ds_Order>1467310037</Ds_Order>
<Ds_Signature>Xsj3sTYPOXtTO+eWogyLs1RxG5UI9VvZAwRxB-
7AQ8fY=</Ds_Signature>
<Ds_MerchantCode>999008881</Ds_MerchantCode>
<Ds_Terminal>871</Ds_Terminal>
<Ds_Response>0000</Ds_Response>
<Ds_AuthorisationCode>446616</Ds_AuthorisationCode>
<Ds_TransactionType>0</Ds_TransactionType>
<Ds_SecurePayment>0</Ds_SecurePayment>
<Ds_Language>1</Ds_Language>
<Ds_ExpiryDate>****</Ds_ExpiryDate>
<Ds_Merchant_Identifier>f30e9f8196cfa0616705fd5ebef-
39d9ab4ee5f38
</Ds_Merchant_Identifier>
<Ds_MerchantData></Ds_MerchantData>
<Ds_Card_Country>724</Ds_Card_Country>
</OPERACION>

</RETORNOXML>

6.
Virtual POS administration module

6.1 Access

The Banco Sabadell Virtual POS includes access to an administration module of operations performed. Access to the intranet is effected via a webpage and offers endless advantages for your merchant management.

The administration module offers real time control of all sales.

In addition to viewing the operations performed, you can always have control over account closes when needed, process the refund of incorrect payments and display the transactions which have not been correctly finalised obtaining information on the error or reason for rejection.

You can access the merchant's administration module at the following web addresses:

- Test environment: https://sis-t.REDSYS.es:25443/canales/bsabadell
- Real environment: https://sis.REDSYS.es/canales/bsabadell

It is advisable to use the **Internet Explorer** browser to enter as some functionalities are only compatible with this browser.

A page will appear for you to enter the user code and administrator password provided earlier by Banco Sabadell, together with the language in which you wish to operate with the administration module.

6.2 Users

Procedures related to registration of new users and modification of access profiles can be carried out in the "Users" section of the Virtual POS administration module. You can also change your password for another which is easy to remember or which you consider more secure.

Two different profiles can be assigned to new users upon registration:

1. **Informative profile**: only transactions and totals can be consulted.
2. **Administrator profile**: in addition to transaction and totals queries it is possible to perform returns, in whole or in part, of sales operations.

The "Users" section of the administration module includes the following options:

1. **Password**: the user access password can be modified.
2. **Users**: all query, registration, de-registration and modification of merchant users can be carried out.
3. **Generate Users**: this enables you to automatically generate, using a merchant code and terminal node, a user to access the administration module with certain default characteristics or permits and to send data to this user to the email address of the specified merchant.

In addition, in accordance with the type of queries allowed for users, the administrator can register two types of users:

1. **Terminal**: to manage the operations performed in the merchant and on a given terminal.
2. **Merchant**: to manage the operations performed by all the terminals of a merchant.

6.3 Operations: queries and administration

The 'Queries" section of the administration module allows you to check the details of operations authorised or rejected by your merchant over the past **365 calendar days** To do so, enter a start and end date for the period to be queried to locate an operation.

Queries concerning operations in the administration module are restricted to 1-month periods. If you need to check longer periods, you must perform consecutive queries of 31-day periods.

For greater search speed, is you know the reference number of the operation you can enter it and access the details immediately.

When you have entered the search parameters and pressed the ACCEPT button a screen appears with a list of the operations matching the search criteria.

The search result, in addition to being displayed on the screen, can be PRINTED or EXPORTED to a text file with separator delimited fields ";".

The response codes shown in the "Result Authorisation No or response code" field, both for operations approved and rejected, match those defined in Annex III – Response Code Table.

6.4 Refund of operations

The Virtual POS administration module allows the merchant to check and generate total or partial refunds of operations which have been processed.

Only those users accessing the administration module with an administrator profile password are authorised to carry out refunds.

Returns of operations over the last 365 calendar days can be carried out.

Para realizar una devolución parcial o total de la operación seleccionada, se deberá pulsar el botón rojo de la columna "Generar devolución" que corresponda a la operación deseada y, a continuación, aparecerá una página para introducir el importe de la devolución. El importe de la devolución no deberá sobrepasar nunca el importe de la operación original y debe ser tecleada siempre con decimales.

In the case of DCC operations (Dynamic Currency Conversion) or Multicurrency operations, the amount must be entered in the currency of the terminal.

When the refund is accepted, a ticket refund page will be shown which can be printed or filed if desired.

Those merchants which carry out pre-authorisation, pre-authentication or deferred pre-authorisation operations can generate confirmations and cancellations of same from the Virtual POS administration module.

6.5 Checking totals

The Virtual POS administration module enables the merchant to check the totals processed.

By pressing the 'Totals' button on the left of the homepage a list of the last 45 sessions appears. Select the desired sessions and pres 'Accept'.

A screen the will appear with the total amounts and number of operations for the session selected.

There exists the option of checking the totals with Breakdown (by card brand) or without Breakdown (360 last sessions).

7.
Installation

This Virtual POS manual provides the necessary information for you or your IT department to install the Virtual POS on your virtual store's website. The installation is simple and basically consists of entering computer instructions in the website which remotely run the Virtual POS software resident on the secure server of Banco Sabadell.

It is advisable this installation be carried out by those persons who regularly perform maintenance on the website and prior to using the Virtual POS in a real environment conduct the necessary tests in a Virtual POS test environment.

The Banco Sabadell Virtual POS accepts different types of processing of merchant operations. Each type entails a different system configuration.

– **Gateway 'realizarPago'**: This is a connection in 'HTML' language and is the common type in processing operations. It is used when it is unnecessary for the merchant to have access to card details. However, this entry also allows card data to be sent. The installation in the merchant's IT system is simple and consists of entering the instructions for the card transactions to be executed via the Virtual POS on the secure servers of Banco Sabadell.

– **Gateway 'WebService'**: The WebService Virtual POS is a product which allows merchants to integrate the Virtual POS within their own Website application. This operation does not accept merchants which have means of payment with authentication of the holder by the card issuing entity.

7.1 Gateway 'realizarPago'

Programming language	**HTML**
Allows holder authentication (3D Secure)	**YES**
Allows the merchant to capture card details	**NO**

The payment page of the merchant's webpage must include a button for the purchaser to identify it with the type of card payment.

When the customer presses the payment button, the merchant fills in a web form with the transaction details whose detailed technical description appears in Annex I' of this manual, and sends it to the following address:

» Test environment:: https://sis-t.redsys.es:25443/sis/realizarPago

» Real environment: https://sis.redsys.es/sis/realizarPago

The payment form must always be shown in a different window displaying the above url so the buyer can identify that they are in the Banco Sabadell payment environment.

The window or frame where the Virtual POS opens must have scroll bars to adapt to the different authentication pages they holder is shown during subsequent processes.

The cardholder who wishes to make the purchase enters the card details directly into the Virtual POS located on the secure Banco Sabadell page; the merchant therefore has no access to the purchaser's card details.

The message has an additional field where the chief purchase details are transmitted securely via the Hash SHA-256 algorithm. (See section 7.6).

Applications to be installed:

REQUIRED: A payment form implemented on the merchant's website.

REQUIRED:
Hash SHA-256 algorithm implemented on the merchant's Internet server.

OPTIONAL: Program to receive and process the online response to the request for payment authorisation.

7.1.1 Continuity of the browser session

Once the cardholder finalises the payment process and is shown the screen with the result, this screen must include the "Close" button for the purchaser to return to the merchant's website session.

The way in which the merchant session with the customer continues will depend on the instructions associated with the 'Close" button. These instructions, about which the merchant owner will have informed Banco Sabadell in the questionnaire effected to commence the registration process, may be:

– **"CLOSE WINDOW" instruction**: On selecting 'Close the window or frame with the payment result will close and the session continued on the merchant page which was in the background.

– **"URL_OK and URL_KO" instruction**: On selecting 'Close the browser session will continue in the same payment page window, rerouting to an URL of which the merchant will first inform Banco Sabadell. This URL may be different if the payment has been authorised (URL_OK) or rejected (URL_KO).

Take into account that if the purchaser closes the browser window, the URL_OK/ URL_KO will not be operative and the session will continue on the merchant page which was in the background.

– **Option for merchants with ONLINE RESPONSE via URL**: In addition to the two above instructions, for merchants with the ONLINE RESPONSE VIA URL service, session continuity can be performed by the merchant website by closing the payment page upon receiving the online response.

The Virtual POS includes strong verification and control systems to detect possible errors in data entry. If an error occurs on data entry, an error code is generated and the operation is terminated. Depending on the type of error, the message shown to the holder will differ. The detailed technical description of the different error messages is included in "Annex II' of this manual.

7.2 Gateway 'Webservice'

Programming language	**XML**
Allows holder authentication (3D Secure)	**NO**
Allows the merchant to capture card details	**YES**

The WebService Virtual POS allows merchants to integrate the Virtual POS within their own Website application. This type of connection to the Virtual POS via the WebService is not allowed in merchants with secure payment methods which request authentication of the holder by the card issuing entity.

The connection to the Virtual POS will be made by sending a requisition via WebService to the following address:

» Test environment: https://sis-t.redsys.es:25443/sis/services/SerClsWSEntrada

» Real environment: https://sis.redsys.es/sis/services/SerClsWSEntrada

The payment gateway will interpret this requisition and perform the relevant checks so as to process the operation. Depending on the result of the operation, a response XML document is created with the result.

To make the purchase via the WebService Virtual POS it is necessary to exchange a series of data both in the requisition messages and the response messages.

7.2.1 Requisition Messages

We can differentiate between three types of requisitions:

– Payment requisitions (card data sent)

– Confirmation requisitions

– Refund requisitions

In each type the message structure and parameters sent/received vary. Below we explain each type with the necessary parameters and show an example of each case.

Payment requisition message

The data which must be included in the message sent to the WebService Virtual POS in XML format a payment requisition and its technical characteristics are described in Annex I of this manual.

Below is a detailed example of how to use these data in payment requisition messages.

```
<REQUEST>
    <DATOSENTRADA>
        <DS_MERCHANT_AMOUNT>145</DS_MERCHANT_
        AMOUNT>
        <DS_MERCHANT_ORDER>151029142229</DS_
        MERCHANT_ORDER>
        <DS_MERCHANT_MERCHANTCODE>327234688</
        DS_MERCHANT_MERCHANTCODE>
        <DS_MERCHANT_CURRENCY>978</DS_MERCHANT_
        CURRENCY>
        <DS_MERCHANT_PAN>4548812049400004</DS_
        MERCHANT_PAN>
        <DS_MERCHANT_EXPIRYDATE>1512</DS_
MERCHANT_
        EXPIRYDATE>
        <DS_MERCHANT_CVV2>285</DS_MERCHANT_CVV2>
        <DS_MERCHANT_TRANSACTIONTYPE>A</DS_
        MERCHANT_TRANSACTIONTYPE>
        <DS_MERCHANT_TERMINAL>2</DS_MERCHANT_
        TERMINAL>
        </DATOSENTRADA>
        <DS_SIGNATUREVERSION>HMAC_SHA256_V1</
        DS_SIGNATUREVERSION>
        <DS_
SIGNATURE>2YW19YQ8rb/0LLav79Y5L24Yw045Kx
        N5hme27605WxY=</DS_SIGNATURE>
</REQUEST>
```

Confirmation/Refund requisition message

The data which must be included in the message to send to the WebService Virtual POS in XML format a Confirmation/Refund requisition and its technical characteristics are described in Annex I of this manual.

Below is a detailed example of how to use these data in payment requisition messages.

```
<DATOSENTRADA>
    <DS_MERCHANT_AMOUNT>145</DS_MERCHANT_AMOUNT>
    <DS_MERCHANT_ORDER>050911523002</DS_MER-
    CHANT_ORDER>
    <DS_MERCHANT_MERCHANTCODE>999008881</DS_MER-
    CHANT_MERCHANTCODE>
    <DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CUR-
    RENCY>
    <DS_MERCHANT_TRANSACTIONTYPE>3</DS_MERCHANT_
    TRANSACTIONTYPE>
    <DS_MERCHANT_TERMINAL>999</DS_MERCHANT_TER-
    MINAL>
    <DS_MERCHANT_MERCHANTSIGNATURE>df-
    66905b10a848ddfa80b202aedcd6b172533cc0</DS_MER-
    CHANT_MERCHANTSIGNATURE>
</DATOSENTRADA>
```

7.2.2 Response messages

The data and technical characteristics which

will be received in the response message after a requisition to the WebService Virtual POS in XML format are detailed in section 7.7 of this manual.

It also includes an example of how to use these data in the response messages.

Example of response message of the operation

```
<RETORNOXML>
   <CODIGO>0</CODIGO>
   <OPERACION>
      <Ds_Amount>145</Ds_Amount>
      <Ds_Currency>978</Ds_Currency>
      <Ds_Order>151029142229</Ds_Order>
      <Ds_Signature>MRvyhuDEpg4BmzfTdgHKrl5qQ9U5UD-
      2Qe8eDadlZtyE=</Ds_Signature>
      <Ds_MerchantCode>327234688</Ds_MerchantCode>
      <Ds_Terminal>2</Ds_Terminal>
      <Ds_Response>0000</Ds_Response>
      <Ds_AuthorisationCode>185714</Ds_AuthorisationCode>
      <Ds_TransactionType>A</Ds_TransactionType>
      <Ds_SecurePayment>0</Ds_SecurePayment>
      <Ds_Language>1</Ds_Language>
      <Ds_MerchantData></Ds_MerchantData>
      <Ds_Card_Country>724</Ds_Card_Country>
   </OPERACION>
</RETORNOXML>
```

Example of response message of the refund operation

```
<RETORNOXML>
   <CODIGO>0</CODIGO>
   <OPERACION>
      <Ds_Amount>1</Ds_Amount>
      <Ds_Currency>978</Ds_Currency>
      <Ds_Order>1499864401</Ds_Order>
      <Ds_Signature>zSrA+/JqBGO4DwGh4FHMe39IL9QzFnp-
      ziTpO6gbjeIc=</Ds_Signature>
      <Ds_MerchantCode>327234688</Ds_MerchantCode>
      <Ds_Terminal>1</Ds_Terminal>
      <Ds_Response>0900</Ds_Response>
      <Ds_AuthorisationCode>479208</Ds_AuthorisationCode>
      <Ds_TransactionType>3</Ds_TransactionType>
      <Ds_SecurePayment>0</Ds_SecurePayment>
      <Ds_Language>1</Ds_Language>
      <Ds_MerchantData></Ds_MerchantData>
      <Ds_Card_Country>724</Ds_Card_Country>
      <Ds_Card_Brand>1</Ds_Card_Brand>
   </OPERACION>
</RETORNOXML>
```

7.3 Locating errors

The Virtual POS includes strong verification and control systems to detect possible errors in data entry. During the installation of the Virtual POS, when sending the payment form, some of the parameters of the form field may be erroneous. If an error occurs on data entry, an error code is generated and the operation is terminated.

Depending on the type of error, the message shown to the holder will differ. The detailed technical description of the different error messages is included in "Annex II' of this manual. This annex lists the possible error values which may be received in the response from the Virtual POS together with the field affected (if any) and the meaning of each one. It also specifies the error message the customer (purchaser) will see in each of these errors.

To locate the erroneous field, see the source code of the error page and search in the HTML text for the chain "**–SIS**". The xxxx numeric value next to the instruction "**<!–SIS-xxxx:–>**" will indicate the type of error according to the table included in Annex II.

7.4 Design of the hash algorithm in the internet server

Banco Sabadell will provide the merchant with a code to be used to sign the data furnished by it, so as to verify not only the identification of the merchant but also that the data have not been altered at any time. The public algorithm Hash SHA-2 will be used as the security protocol, which guarantees the minimum security requisites as regards authentication of origin.

The same algorithm will be used to assure the merchant of the authenticity of the

response data if the notification URL is supplied by the merchant.

The type of SHA2 code is not available in php versions lower than version 5.0. If your server uses any previous version contact the technical service of Banco Sabadell to find an alternative solution.

The calculation method of the algorithm differs depending on the type of payment selected.

7.4.1 Gateway 'realizarPago'

The payment page of the merchant's webpage must include a button for the purchaser to identify it with the type of card payment.

The button must be associated with the hidden payment form described below. When the buyer selects this button, the merchant must send the operation payment form to the Banco Sabadell server.

The payment form must always be shown in a different window displaying the above url so the buyer can identify that they are in the Banco Sabadell payment environment.

For CES merchants, **the window where the Virtual PoS opens must have vertical and horizontal scroll bars** to adapt to the different authentication pages the holder is shown during subsequent processes.

Below are the data to be included in the payment form:

DATUM	NAME OF FIELD	COMMENTS
Signature version:	Ds_SignatureVersion	Constant indicating the signature version being used.
Details of the operation	Ds_MerchantParameters	Chain in JSON format with all the parameters of the request encoded in Base 64
Signature	Ds_Signature	Result of the HMAC SHA256 of the encoded JSON chain in Base 64 sent in the previous parameter.

To create the **Ds_MerchantParameters** field, all the fields marked **as required** in the table below must be used. The remaining fields are optional and may be included if the merchant wishes.

DATUM	NAME OF FIELD	LENGTH	COMMENTS
Merchant number Fuc code	Ds_Merchant_ MerchantCode	9 N	**Required**. Fixed code assigned by Banco Sabadell.
Terminal number	Ds_Merchant_ Terminal	3 N	**Required**. Standard: 1 - Operations in euros (Ds_Merchant_Currency= 978) If more terminals are necessary, contact the Banco Sabadell technical service Terminal number assigned by bank. Three is considered the maximum length
Order number	Ds_Merchant_ Order	Min. 4N Max. 12 AN For “Card on File” in field must be max. 10 positions As the Virtual PoS will add 2 more positions indicating the payment order number.	**Required**. The first 4 digits must be numerical; the remaining digits can only use the following ASCII characters From 30 = 0 to 39 = 9 From 65 = A to 90 = Z From 97 = a to 122 = z The code must be different from previous transactions.
Amount	Ds_Merchant_ Amount	12 N	**Required**. The last two positions are considered decimals, except in Yen.

Currency	Ds_Merchant_ Currency	4 N	**Required**. 978 – EURO 840 – USD 826 – GBP 392 – JPY 756 – CHF 124 – CAD 4 is considered the maximum length
Type of transaction	Ds_Merchant_ Transaction Type	1 N	**Required**. 0 - Standard payment 1 - Pre-authorisation 2 - Confirmation of pre-authorisation 3 - Partial or full refund 7 - Authentication 8 - Confirmation of authentication 9 - Cancellation of pre-authorisation L - Card in Initial File M - Successive Card on File O - Deferred pre-authorisation P - Confirmation of Deferred Pre-authorisation Q - Cancellation of Deferred Pre-authorisation
Product Description	Ds_Merchant_ Product Description	Max. 125 AN	**Required.** This field is shown to the holder on the purchase confirmation screen.
Name and surnames of holder	Ds_Merchant_ Titular	Max. 60 AN	**Required.** This field is shown to the holder on the purchase confirmation screen.
URL	Ds_Merchant_ MerchantURL	250 AN	**Required if merchant has online notification**. URL of merchant which will receive a post with the transaction data.

URLOK	Ds_Merchant_ UrlOK	250 AN	**Optional**. If sent it will be used as URLOK, ignoring that configured in the administration module if any.
URLKO	Ds_Merchant_ UrlKO	250 AN	**Optional**. If sent it will be used as URLKO, ignoring that configured in the administration module if any.
Name of merchant	Ds_Merchant_ MerchantName	25 AN	**Optional**. Name of merchant appearing on customer payment page, if any.
Holder's language	Ds_Merchant_ Consumer Language	3 N	**Optional**. 0 – Customer 1 – Spanish 2 – English 3 – Catalan 4 – French 5 – German 6 – Dutch 7 – Italian 8 – Swedish 9 – Portuguese 10 – Valencian 11 – Polish 12 – Galician 13 – Basque
Merchant data	Ds_Merchant_ MerchantData	1024 AN	**Optional**. Free information of merchant to be received in online response (via URL or e-mail).
Authorisation code	Ds_Merchant_ Authorisation Code	6 N	**Optional**.

Identifier	Ds_Merchant_ Identifier	Max. 40 AN	**Exclusive field for payment by reference.** Value of the field is Required for first payment transaction. For subsequent payments, the value will be the identifier that the Bank has sent in the first payment response message.
Group of merchants	Ds_Merchant_ Group	Max. 9 N	**Exclusive field for payment by reference. Optional**. Allows to associate an identifier to a set of merchants.
Additional screens	Ds_Merchant_ DirectPayment	'True' or 'false'	**Exclusive field for payment by reference. Optional**. This parameter acts as a flag to indicate if additional screens must be shown (DCC, Splitting, Authentication, etc.).

Example of request dispatch form

Below is an example of the payment request form:

Example of payment form **without sending** the card details:

```
<form id="form1" method="post" action="https://sis-t.
  redsys.es:25443/sis/realizarPago" target="_blank">
<input type="hidden" name="Ds_SignatureVersion"
  value="HMAC_SHA256_V1" />
<input type="hidden" name="Ds_MerchantParameters"
  value="eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiL-
  CJEU19NRVJDSEFOVF9PUkRFUiI6MTQ2Mjc5NjIwMiwiR-
  FNfTUVSQ0hBTlRfTUVSQ0hBTlRDT0RFIjoiMzI3MjM0N-
  jg4IiwiRFNfTUVSQ0hBTlRfQ1VSUkVOQ1kiOiI5NzgiL-
  CJEU19NRVJDSEFOVF9UUkFOU0FDVEIPTlRZUEUiOiI-
  wIiwiRFNfTUVSQ0hBTlRfVEVSTUlOQUwiOiIxIiwiRFNfTU-
  VSQ0hBTlRfTUVSQ0hBTlRVUkwiOiJodHRwOlwvXC93d-
  3cud2ViZGVsY29tZXJjaW8uY29tXC91cmxkZW5vdGl-
  maWNhY2lvbi5waHAiLCJEU19NRVJDSEFOVF9VUkxPSy-
  I6Imh0dHA6XC9cL3d3dy53ZWJkZWxjb21lcmNpby5jb-
  21cL3VybG9rLnBocCIsIkRTX01FUkNlQU5UX1VSTEtPI-
  joiaHR0cDpcL1wvd3d3LndlYmRlbGNvbWVyY2lvLmNv-
  VwvdXJsa28ucGhwIn0=" />
<input type="hidden" name="Ds_Signature" value="arb-
  jAnswMyybenZlBKqxS8Fdw4nSWRdRXfmTPhHZkJg=" />
</form>
```

Example of payment form **sending** the card details:

```
<form id="form1" method="post" action="https://sis-t.
  redsys.es:25443/sis/realizarPago" target="_blank">
<input type="hidden" name="Ds_SignatureVersion"
  value="HMAC_SHA256_V1" />
<input type="hidden" name="Ds_MerchantParameters"
  value="eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJE-
  U19NRVJDSEFOVF9PUkRFUiI6MTQ2Mjc5NTk1MSwiR-
  FNfTUVSQ0hBTlRfTUVSQ0hBTlRDT0RFIjoiMzI3MjM0N-
  jg4IiwiRFNfTUVSQ0hBTlRfQ1VSUkVOQ1kiOiI5NzgiL-
  CJEU19NRVJDSEFOVF9UUkFOU0FDVEIPTlRZUEUiOiI-
  wIiwiRFNfTUVSQ0hBTlRfVEVSTUlOQUwiOiIxIiwiRFNfTU-
  VSQ0hBTlRfTUVSQ0hBTlRVUkwiOiJodHRwOlwvXC93d-
  3cud2ViZGVsY29tZXJjaW8uY29tXC91cmxkZW5vdGl-
  maWNhY2lvbi5waHAiLCJEU19NRVJDSEFOVF9VUkxPSy-
  I6Imh0dHA6XC9cL3d3dy53ZWJkZWxjb21lcmNpby5jb-
  21cL3VybG9rLnBocCIsIkRTX01FUkNlQU5UX1VSTEtPI-
  joiaHR0cDpcL1wvd3d3LndlYmRlbGNvbWVyY2lvLmN-
  vbVwvdXJsa28ucGhwIiwiRFNfTUVSQ0hBTlRfUEFOIjoiN-
  DU0ODgxMjA0OTQwMDAwNCIsIkRTX01FUkNlQU5UX-
  0VYUEISWURBVEUiOilxNjEyliwiRFNfTUVSQ0hBTlRfQ-
  1ZWMiI6IjEyMyJ9" />
<input type="hidden" name="Ds_Signature" value="wLir-
  BHcT3mc01WmwuE2/qy1cL6o46D+eJjlp5dih/40=" />
</form>
```

To facilitate the integration of the merchant, below is a detailed explanation of how to assemble the payment request message.

7.4.1.1 Identifying the version of the signature algorithm to be used

It is necessary to identify in the request the specific version of algorithm being used for the signature. The value **HMAC_SHA256_V1** is currently used to identify the version of all requests so this will be the value of the **Ds_SignatureVersion** parameter as can be seen in the example form above.

7.4.1.2 Generating the request data chain

A chain with all the request data must be generated in JSON format. JSON is an open, text-based data exchange format. As with XML, it is designed to be legible and independent of the technological platform. The encoding of data in JSON is very light, making it idea for exchanging data in Web applications.

The name of each parameter must be indicated in capital letters or with a "CamelCase" structure (For example: DS_MERCHANT_AMOUNT or Ds_Merchant_Amount). The list of parameters that can be included in the request is described in section **6.1 Merchant website payment form**. Below is an example of the JSON object of a request:

Example of card data **without sending**:

{"DS_MERCHANT_AMOUNT":"145","DS_MERCHANT_ORDER":1462796202,"DS_MERCHANT_MERCHANTCODE":"327234688","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MERCHANT_TERMINAL":"1","DS_MERCHANT_MERCHANTURL":"http:\/\/www.webdelcomercio.com\/urldenotificacion.php","DS_MERCHANT_URLOK":"http:\/\/www.webdelcomercio.com\/urlok.php","DS_MERCHANT_URLKO":"http:\/\/www.webdelcomercio.com\/urlko.php"}

Example of card data **with sending**:

{"DS_MERCHANT_AMOUNT":"145","DS_MERCHANT_ORDER":1462795951,"DS_MERCHANT_MERCHANTCODE":"327234688","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MERCHANT_TERMINAL":"1","DS_MERCHANT_MERCHANTURL":"http:\/\/www.webdelcomercio.com\/urldenotificacion.php","DS_MERCHANT_URLOK":"http:\/\/www.webdelcomercio.com\/urlok.php","DS_MERCHANT_URLKO":"http:\/\/www.webdelcomercio.com\/urlko.php","DS_MERCHANT_PAN":"4548812049400004","DS_MERCHANT_EXPIRYDATE":"1612","DS_MERCHANT_CVV2":"533"}

Once the JSON chain is assembled with all the fields it must be encoded in BASE64 without hard returns to ensure it remains constant and is not altered when passing through the customer/buyer browser.

Below is the JSON object just shown encoded in BASE64:

Example of JSON encoded **without sending** card data:

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6MTQ2Mjc5NjIwMiwiRFNfTUVSQ0hBTlRfTUVSQ0hBTlRDT0RFIjoiMzI3MjM0Njg4IiwiRFNfTUVSQ0hBTlRfQ1VSUkVOQ1kiOiI5NzgiLCJEU19NRVJDSEFOVF9UUkFOU0FDVElPTlRZUEUiOiIwIiwiRFNfTUVSQ0hBTlRfVEVSTUlOQUwiOiIxIiwiRFNfTUVSQ0hBTlRfTUVSQ0hBTlRVUkwiOiJodHRwOlwvXC93d3cud2ViZGVsY29tZXJjaW8uY29tXC91cmxkZW5vdGlmaWNhY2lvbi5waHAiLCJEU19NRVJDSEFOVF9VUkxPSyI6Imh0dHA6XC9cL3d3dy53ZWJkZWxjb21lcmNpby5jb21cL3VybG9rLnBocCIsIkRTX01FUkNIQU5UX1VSTEtPIjoiaHR0cDpcL1wvd3d3LndlYmRlbGNvbWVyY2lvLmNvbVwvdXJsa28ucGhwIn0=

Example of card data **with sending** of card data:

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6MTQ2Mjc5NTk1MSwiRFNfTUVSQ0hBTlRfTUVSQ0hBTlRDT0RFIjoiMzI3MjM0Njg4IiwiRFNfTUVSQ0hBTlRfQ1VSUkVOQ1kiOiI5NzgiLCJEU19NRVJDSEFOVF9UUkFOU0FDVElPTlRZUEUiOiIwIiwiRFNfTUVSQ0hBTlRfVEVSTUlOQUwiOiIxIiwiRFNfTUVSQ0hBTlRfTUVSQ0hBTlRVUkwiOiJodHRwOlwvXC93d3cud2ViZGVsY29tZXJjaW8uY29tXC91cmxkZW5vdGlmaWNhY2lvbi5waHAiLCJEU19NRVJDSEFOVF9VUkxPSyI6Imh0dHA6XC9cL3d3dy53ZWJkZWxjb21lcmNpby5jb21cL3VybG9rLnBocCIsIkRTX01FUkNIQU5UX1VSTEtPIjoiaHR0cDpcL1wvd3d3LndlYmRlbGNvbWVyY2lvLmNvbVwvdXJsa28ucGhwIiwiRFNfTUVSQ0hBTlRfUEFOIjoiNDU0ODgxMjA0OTQwMDAwNCIsIkRTX01FUkNIQU5UX0VYUElSWURBVEUiOiIxNjEyIiwiRFNfTUVSQ0hBTlRfQ1ZWMiI6IjUzMyJ9

The chain created by encoding in BASE64 of the JSON will be the value of the Ds_Mer-

chantParameters parameter, as can be seen in the example of a form shown at the beginning of this section.

7.4.1.3 Identifying the code to be used for signing

To calculate the signature it is necessary to use a specific code for each terminal. The merchant code to be used is that received from Banco Sabadell in an SMS.

IMPORTANT NOTE: This code must be stored in the merchant server as secure as possible to avoid any fraudulent use of same. The merchant is responsible for properly safeguarding and maintaining the code secret.

7.4.1.4 Signing the request data

Once the data chain to be signed has been created and the specific code of the terminal, the signature must be calculated by applying the following steps:

1. A specific code is generated per operation. To obtain the code to be used in an operation 3DES encryption must be made between the merchant code, which must be first decoded in BASE 64, and the value of the operation order number (Ds_Merchant_Order).
2. The HMAC SHA256 of the value of the Ds_MerchantParameters is calculated and the code obtained in the preceding step.
3. The result obtained is encoded in BASE 64, and the result will be the value of the Ds_Signature parameter, as shown in the example of the form at the beginning of section 3.

7.4.1.5 Use of the help libraries

The previous sections described the method for sending the payment request using a connection via the entry **Make payment** and the signature system based on HMAC SHA256. This section explains how the libraries available in PHP, JAVA and .NET are used to facilitate developments and generate payment form fields. The use of the libraries supplied by Banco Sabadell is optional although they simplify the developments to be carried out by the merchant.

7.4.1.5.1 PHP library

Below are the steps to be followed by the merchant to use the PHP library provided by Banco Sabadell:

1. Import the chief library file as shown below:

```
include("./apiRedsys.php");
```

The merchant must decide whether to perform the import using the "include" or "required" function, in accordance with the developments made.

2. Define an object of the chief library class, as shown below:

```
$miObj = new RedsysAPI;
```

3. Calculate the **Ds_MerchantParameters** parameter. To calculate this parameter, first all the parameters of the payment request to be sent must be added.

   Important: There is no specific order to adding parameters, so they may be included in any order as desired.

   Example of parameters **without sending** card details:

```
$miObj->setParameter("DS_MERCHANT_AMOUNT",
$importe);
$miObj->setParameter("DS_MERCHANT_CURRENCY",
$moneda);
$miObj->setParameter("DS_MERCHANT_ORDER",
strval($numPedido));
$miObj->setParameter("DS_MERCHANT_MERCHANT-
CODE", $merchantCode);
$miObj->setParameter("DS_MERCHANT_TERMINAL",
```

```
$terminal);
$miObj->setParameter("DS_MERCHANT_TRANSACTION-
TYPE", $transactionType);
$miObj->setParameter("DS_MERCHANT_MERCHAN-
TURL", $merchantURL);
$miObj->setParameter("DS_MERCHANT_URLOK",
$urlOK);
$miObj->setParameter("DS_MERCHANT_URLKO",
$urlKO);
```

Example of parameters **sending** card details:

```
$miObj->setParameter("DS_MERCHANT_AMOUNT",
$importe);
$miObj->setParameter("DS_MERCHANT_CURRENCY",
$moneda);
$miObj->setParameter("DS_MERCHANT_ORDER",
strval($numPedido));
$miObj->setParameter("DS_MERCHANT_MERCHANT-
CODE", $merchantCode);
$miObj->setParameter("DS_MERCHANT_TERMINAL",
$terminal);
$miObj->setParameter("DS_MERCHANT_TRANSACTION-
TYPE", $transactionType);
$miObj->setParameter("DS_MERCHANT_MERCHAN-
TURL", $merchantURL);
$miObj->setParameter("DS_MERCHANT_URLOK",
$urlOK);
$miObj->setParameter("DS_MERCHANT_URLKO",
$urlKO);
$miObj->setParameter("DS_MERCHANT_PAN", $numTar-
jeta);
$miObj->setParameter("DS_MERCHANT_EXPIRYDATE",
$fechaCaducidad);
$miObj->setParameter("DS_MERCHANT_CVV2", $cvv2);
```

Lastly, to calculate the **Ds_Merchant-Parameters** parameter, it is necessary to call the library function "createMerchantParameters()", as shown below:

```
$params = $miObj->createMerchantParameters();
```

4. Calculate the **Ds_Signature** parameter To calculate this parameter, call the library "createMerchantSignature()"with the merchant code provided as shown below:

```
$clave = 'sq7HjrU0BfKmC576ILgskD5srU870gJ7';
$firma = $miObj->createMerchantSignature($clave);
```

5. Once the value of the parameters **Ds_MerchantParameters** and **Ds_Signature** have been obtained, complete the payment form with these values as shown below:

```
<form name="form" action="https://sis-t.redsys.
es:25443/sis/realizarPago"
method="POST" target="_blank">
  <input type="hidden" name="Ds_SignatureVersion"
  value="<?php echo $version; ?>"/>
  <input type="hidden" name="Ds_MerchantParame-
  ters" value="<?php echo $params; ?>"/>
  <input type="hidden" name="Ds_Signature"
  value="<?php echo $firma; ?>" />
  <input type="submit" value="Realizar Pago" />
</form>
```

7.4.1.5.2 Java library

Below are the steps to be followed by the merchant to use the JAVA library provided by Banco Sabadell:

1. Import the library as shown below:

```
<%@page import="sis.redsys.api.ApiMacSha256"%>
```

The merchant must include in the project construction the libraries (JARs) provided:

2. Define an object of the chief library class, as shown below:

```
ApiMacSha256 apiMacSha256 = new ApiMacSha256();
```

3. Calculate the **Ds_MerchantParameters** parameter. To calculate this parameter, first all the parameters of the payment request to be sent must be added.

Important: There is no specific order to adding parameters, so they may be included in any order as desired.

Example of parameters **without sending** card details:

```
apiMacSha256.setParameter("DS_MER-
CHANT_AMOUNT", importe);apiMacSha256.
setParameter("DS_MERCHANT_ORDER", numPe-
dido);apiMacSha256.setParameter("DS_MERCHANT_
MERCHANTCODE", merchantCode);
apiMacSha256.setParameter("DS_MERCHANT_CUR-
RENCY", moneda);
apiMacSha256.setParameter("DS_MERCHANT_
TRANSACTIONTYPE", transactionType);apiMacSha256.
setParameter("DS_MERCHANT_TERMINAL", terminal);
apiMacSha256.setParameter("DS_MERCHANT_MER-
CHANTURL", merchantURL);
apiMacSha256.setParameter("DS_MERCHANT_UR-
LOK", urlOK)
```

```
apiMacSha256.setParameter("DS_MERCHANT_
URLKO", urlKO);
```

Example of parameters **sending** card details:

```
apiMacSha256.setParameter("DS_MERCHANT_
AMOUNT", importe);
apiMacSha256.setParameter("DS_MERCHANT_OR-
DER", numPedido);
apiMacSha256.setParameter("DS_MERCHANT_MER-
CHANTCODE", merchantCode);
apiMacSha256.setParameter("DS_MERCHANT_CUR-
RENCY", moneda);
apiMacSha256.setParameter("DS_MERCHANT_
TRANSACTIONTYPE", transactionType);
apiMacSha256.setParameter("DS_MERCHANT_TERMI-
NAL", terminal);
apiMacSha256.setParameter("DS_MERCHANT_MER-
CHANTURL", merchantURL);
apiMacSha256.setParameter("DS_MERCHANT_UR-
LOK", urlOK);
apiMacSha256.setParameter("DS_MERCHANT_
URLKO", urlKO);
apiMacSha256.setParameter("DS_MERCHANT_PAN",
numTarjeta);
apiMacSha256.setParameter("DS_MERCHANT_EX-
PIRYDATE", expiryDate);
apiMacSha256.setParameter("DS_MERCHANT_CVV2",
cvv2);
```

Lastly, call the library function "createMerchantParameters()", as shown below:

```
String params = apiMacSha256.createMerchantPa-
rameters();
```

4. Calculate the **Ds_Signature** parameter To calculate this parameter call the "createMerchantSignature()" library function with the code of the merchant provided as shown below

```
String clave = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
String firma = apiMacSha256.createMerchantSigna-
ture(clave);
```

5. Once the value of the parameters **Ds_MerchantParameters** and **Ds_Signature** have been obtained, complete the payment form with these values as shown below:

```
<form action="https://sis-t.redsys.es:25443/sis/
realizarPago"
method="POST" target="_blank">
<input type="hidden" name="Ds_SignatureVersion"
value="HMAC_SHA256_V1" />
<input type="hidden" name="Ds_MerchantParameters"
value="<%= params %>" />
<input type="hidden" name="Ds_Signature"
value="<%= firma %>" />
<input type="submit" value="Realizar Pago" />
</form>
```

7.4.1.5.3 .NET library

Below are the steps to be followed by the merchant to use the .NET library provided by Redsys:

1. Import the RedsysAPI and Newronsoft library Json into your project.

2. Calculate the **Ds_MerchantParameters** parameter. To calculate this parameter, first all the parameters of the payment request to be sent must be added.

   Important: There is no specific order to adding parameters, so they may be included in any order as desired.

   Example of parameters **without sending** card details:

```
//Creación del objeto
RedsysAPI r = new RedsysAPI();

r.SetParameter("DS_MERCHANT_AMOUNT",amount);
r.SetParameter("DS_MERCHANT_ORDER",order);
r.SetParameter("DS_MERCHANT_MERCHANTCODE",mer-
chantCode);
r.SetParameter("DS_MERCHANT_CURRENCY",currency);
r.SetParameter("DS_MERCHANT_TRANSACTION-
TYPE",transactionType);
r.SetParameter("DS_MERCHANT_TERMINAL",terminal);
r.SetParameter("DS_MERCHANT_MERCHANTURL",mer-
chantURL);
r.SetParameter("DS_MERCHANT_URLOK",urlOK);
r.SetParameter("DS_MERCHANT_URLKO",urlKO);
```

   Example of parameters **sending** card details:

```
r.SetParameter("DS_MERCHANT_AMOUNT",amount);
r.SetParameter("DS_MERCHANT_ORDER",order);
r.SetParameter("DS_MERCHANT_MERCHANTCODE",mer-
chantCode);
r.SetParameter("DS_MERCHANT_CURRENCY",currency);
r.SetParameter("DS_MERCHANT_TRANSACTION-
TYPE",transactionType);
r.SetParameter("DS_MERCHANT_TERMINAL",terminal);
r.SetParameter("DS_MERCHANT_MERCHANTURL",mer-
chantURL);
r.SetParameter("DS_MERCHANT_URLOK",urlOK);
r.SetParameter("DS_MERCHANT_URLKO",urlKO);
r.SetParameter("DS_MERCHANT_PAN",pan);
r.SetParameter("DS_MERCHANT_EXPIRYDATE",fecha);
r.SetParameter("DS_MERCHANT_CVV2",cvv2);
```

Lastly, call the library function "createMerchantParameters()", as shown below:

```
string parms = r.createMerchantParameters();
Ds_MerchantParameters.Value = parms;
```

3. Calculate the **Ds_Signature** parameter To calculate this parameter, call the library function "createMerchantSignature()"with the merchant code provided as shown below:

```
string parms = r.createMerchantParameters();
Ds_MerchantParameters.Value = parms;
```

4. Once the value of the parameters **Ds_MerchantParameters** and **Ds_Signature** have been obtained, complete the payment form with these values as shown below:

```
<form action=" https://sis-t.redsys.es:25443/sis/
realizarPago" method="post">
<input runat="server" type="text" id="Ds_Signature-
Version"
name="Ds_SignatureVersion" value="" />
<input runat="server" type="hidden" id="Ds_Merchant-
Parameters"
name="Ds_MerchantParameters" value="" />
<input runat="server" type="hidden" id="Ds_Signature"
name="Ds_Signature" value="" />
<input id="Submit1" runat="server" type="submit"
value="Realizar Pago" />
</form>
```

7.4.1.6 Reception of the online notification

The online notification is an optional function that enables the web store to receive the result of the transaction online and in real time once the buyer has completed the process on the Virtual PoS.

The merchant must capture and **verify all the parameters together with the signature** of the online notification before any execution on its server.

The use of the help libraries provided by Banco Sabadell is set out in the following sub-sections and will depend on the type of notification configured.

7.4.1.6.1 Synchronous and asynchronous synchronisation

The previous sections described the method for creating payment requests using a connection via the entry **Make payment** and the signature system based on HMAC SHA256. This section explains how the libraries available in PHP, JAVA and .NET **are used to facilitate developments for the online reception of the notification parameters and validation of the signature**. The use of the libraries supplied by Banco Sabadell is optional although they simplify the developments to be carried out by the merchant.

For correct reception of the online notification by the merchant server the following requirements must be met:

- The url must be accessible via the internet
- It must not request the username and password
- It must not reroute to third-party pages
- It must be prepared to receive the parameters via POST.

It is possible that due to security matters you wish to limit access to the server so that only authorised connections are made. If this is the case, below are the IP's of the notification servers from which the online communications will be made:

195.76.9.117
195.76.9.149
193.16.243.13
193.16.243.173
195.76.9.187
195.76.9.222
194.224.159.47
194.224.159.57

7.4.1.6.1.1 PHP library

Below are the steps to be followed by the merchant to use the PHP library provided by Banco Sabadell:

1. Import the chief library file as shown below:

```
include("./apiRedsys.php");
```

   The merchant must decide whether to perform the import using the "include" or "required" function, in accordance with the developments made.

2. Define an object of the chief library class, as shown below:

```
$miObj = new RedsysAPI;
```

3. Capture the parameters of the online notification:

```
$version = $_POST["Ds_SignatureVersion"];
$params = $_POST["Ds_MerchantParameters"];
$firmaRecibida = $_POST["Ds_Signature"];
```

4. Validate the **Ds_Signature** parameter To validate this parameter, it is necessary to calculate the signature and compare it with the **Ds_Signature** parameter captured. To do so, call the function of the "createMerchantSignatureNotif()" library using the merchant code provided and the **Ds_MerchantParameters** parameter captured as shown below:

```
$clave = 'sq7HjrU0BfKmC576ILgskD5srU870gJ7';
$firmaCalculada = $miObj->createMerchantSignatureNo
tif($clave,$params);
```

   Upon completion, it is now possible to check whether the signature sent matches the value of the signature calculated, as shown below:

```
if ($firmaCalculada === $firmaRecibida)
{
//FIRMA OK. Realizar tareas de servidor.
}
else
{
//FIRMA KO. Error, firma inválida.
}
```

   Once the "createMerchantSignatureNotif()" function has been called up, the value of any parameter susceptible for inclusion in the online notification can be obtained, as shown in section 6.3 Online response. To obtain the value of a parameter we must call the "getParameter()" function of the library with the parameter name as shown below to obtain the response code:

```
$codigoResp = $miObj->getParameter("Ds_Response");
```

   IMPORTANT NOTE: To ensure the security and source of the notifications the merchant must validate the signature received and all the parameters sent in the notification

7.4.1.6.1.2 Java library

Below are the steps to be followed by the merchant to use the JAVA library provided by Banco Sabadell:

1. Import the library as shown below:

```
<%@page import="sis.redsys.api.ApiMacSha256"%>
```

   The merchant must include in the project construction all the libraries (JARs) provided:

2. Define an object of the chief library class, as shown below:

```
ApiMacSha256 apiMacSha256 = new ApiMacSha256();
```

3. Capture the parameters of the online notification:

```
String version = request.getParameter("Ds_Signature-
Version");
String params = request.getParameter("Ds_MerchantPa-
rameters");
String signatureRecibida = request.getParameter("Ds_
Signature");
```

4. Validate the **Ds_Signature** parameter To validate this parameters, it is necessary to calculate the signature and compare

it with the **Ds_Signature** parameter captured. To do so, call the function of the "createMerchantSignatureNotif f()" library using the merchant code provided and the **Ds_MerchantParameters** parameter captured as shown below:

```
String clave = "sq7HjrU0BfKmC576ILgskD5srU870gJ7";
String signatureCalculada = apiMacSha256.createMerch
antSignatureNotif(clave, params);
```

Upon completion, it is now possible to check whether the signature sent matches the value of the signature calculated, as shown below:

```
if (signatureCalculada.equals(signatureRecibida)) {
  System.out.println("FIRMA OK. Realizar tareas en el
  servidor");
} else {
  System.out.println("FIRMA KO. Error, firma inválida");
}
```

Once the "**createMerchantSignatureNotif**()" function has been called up, the value of any parameter susceptible for inclusion in the online notification can be obtained, as shown in section 6.3 **Online response**. To obtain the value of a parameter we must call the "**getParameter**()" function of the library with the parameter name as shown below to obtain the response code:

```
String codigoRespuesta = apiMacSha256.getParame-
ter("DS_Response");
```

IMPORTANT NOTE: To ensure the security and source of the notifications the merchant must validate the signature received and all the parameters sent in the notification.

7.4.1.6.1.3 .NET library

Below are the steps to be followed by the merchant to use the JAVA library provided by Redsys:

1. Import the RedsysAPI and Newronsoft library JSON into your project.
2. Capture the parameters of the online notification:

```
//Creación del objeto
RedsysAPI r = new RedsysAPI();

// Obtener la variable Ds_SignatureVersion vía POST
if (Request.Form["Ds_SignatureVersion"] != null)
{
      version = Request.Form["Ds_SignatureVersion"];
}
// Obtener la variable Ds_MerchantParameters vía POST
if (Request.Form["Ds_MerchantParameters"] != null)
{
      parms = Request.Form["Ds_MerchantParameters"];
}
// Obtener la variable Ds_Signature vía POST
if (Request.Form["Ds_Signature"] != null)
{
      firmaRecibida = Request.Form["Ds_Signature"];
}
```

IMPORTANT NOTE: To ensure the security and source of the notifications the merchant must validate the signature received and all the parameters sent in the notification.

3. Validate the **Ds_Signature** parameter To validate this parameters, it is necessary to calculate the signature and compare it with the **Ds_Signature** parameter captured. To do so, call the function of the "createMerchantSignatureNotif()" library using the merchant code provided and the **Ds_MerchantParameters** parameter captured as shown below:

```
string clave = "sq7HjrU0BfKmC576ILgskD5srU870gJ7";
string notif = r.createMerchantSignatureNotif(clave, data);
```

Upon completion, it is now possible to check whether the signature sent matches the value of the signature calculated, as shown below:

```
if (notif.Equals(firmaRecibida) && notif != "")
{
//FIRMA OK. Realizar tareas de servidor
}
else
{
//FIRMA KO. Error, firma inválida.
}
```

IMPORTANT NOTE: To ensure the security and source of the notifications the merchant must validate the signature

received and all the parameters sent in the notification.

7.4.1.7 Return of browsing control

Once the cardholder finalises the payment process and is shown the screen with the result, this screen must include the "Close" button for the purchaser to return to the merchant's website session.

The way in which the merchant session with the customer continues will depend on the instructions associated with the 'Close" button.

These instructions may be:

"CLOSE WINDOW" instruction: on selecting 'Close" the window with the payment result will close and the session continues on the merchant page which remained in the background.

"URL_OK" and "URL_KO" instructions: On selecting 'Close the browser session will continue in the same payment page window, rerouting to an URL of which the merchant will first inform Banco Sabadell. This URL may be different if the payment has been authorised (URL_OK) or rejected (URL_KO).

The merchant must capture and verify, if the merchant has activated the return of the parameters of the operation via the URL, the browsing control parameters before any execution on its server, although it is advisable not to perform any action on the server via these URL as the customer may modify the response values.

The use of the help libraries provided by Banco Sabadell to capture and validate the browsing control return parameters is set out below.

7.4.1.7.1 Use of the help libraries

The sections above describe the type of access to SIS using the connection via **Make Payment**. This section explains how the libraries available in PHP, JAVA and .NET to to facilitate developments for the online reception of the browsing control return parameters. The use of the libraries supplied by Banco Sabadell is optional although they simplify the developments to be carried out by the merchant.

7.4.1.7.1.1 PHP library

Below are the steps to be followed by the merchant to use the PHP library provided by Banco Sabadell:

1. Import the chief library file as shown below:

   ```
   include("./apiRedsys.php");
   ```

   The merchant must decide whether to perform the import using the "include" or "required" function, in accordance with the developments made.

2. Define an object of the chief library class, as shown below:

   ```
   $miObj = new RedsysAPI;
   ```

3. Capture the parameters of the online notification:

   ```
   $version = $_GET["Ds_SignatureVersion"];
   $params = $_GET["Ds_MerchantParameters"];
   $signatureRecibida = $_GET["Ds_Signature"];
   ```

IMPORTANT NOTE: It is important to validate all the parameters sent in the communication. To update the order status online this communication must NOT be used but the online notification described in the other sections, as the browsing return depends on the client actions in the browser.

4. Validate the **Ds_Signature** parameter. To validate this parameters, it is necessary to calculate the signature and compare it with the **Ds_Signature** parameter captured. To do so, call the function of the "createMerchantSignatureNotif()" library using the merchant code provided and the **Ds_MerchantParameters** parameter captured as shown below:

```
$clave = 'sq7HjrU0BfkmC576ILgskD5srU870gJ7' ;
$signatureCalculada = $miObj->createMerchantSignatureNotif($clave,$params);
```

Upon completion, it is now possible to check whether the signature sent matches the value of the signature calculated, as shown below:

```
if ($signatureCalculada ===$signatureRecibida)
{
    //FIRMA OK. Realizar tareas de servidor.
}

else
{
    //FIRMA KO. Error, firma inválida.
}
```

Once the "decodeMerchantParameters()" function has been called up, the value of any parameter susceptible for inclusion in the browsing control return can be obtained, as shown in section **Online response**. To obtain the value of a parameter we must call the "getParameter()" function of the library with the parameter name as shown below to obtain the response code:

```
$codigoRespuesta = $miObj->getParameter("Ds_Response");
```

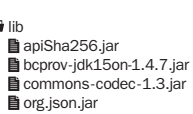
7.4.1.7.1.2 Java library

Below are the steps to be followed by the merchant to use the JAVA library provided by Banco Sabadell:

1. Import the library as shown below:

```
<%@page import="sis.redsys.api.ApiMacSha256"%>
```

The merchant must include in the project construction all the libraries (JARs) provided:



2. Define an object of the chief library class, as shown below:

```
ApiMacSha256 apiMacSha256 = new ApiMacSha256();
```

3. Capture the browsing control return parameters:

```
String version = request.getParameter("Ds_SignatureVersion");
String params = request.getParameter("Ds_MerchantParameters");
String signatureRecibida = request.getParameters("Ds_Signature");
```

4. Validate the **Ds_Signature** parameter To validate this parameters, it is necessary to calculate the signature and compare it with the **Ds_Signature** parameter captured. To do so, call the function of the "createMerchantSignatureNotif()" library using the merchant code provided and the **Ds_MerchantParameters** parameter captured as shown below:

```
String clave = "sq7HjrU0BfkmC576ILgskD5srU870gJ7";
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotif(clave, params);
```

Upon completion, it is now possible to check whether the signature sent matches the value of the signature calculated, as shown below:

```
if (signatureCalculada.equals(signatureRecibida))
{
    System.out.println("FIRMA OK. Realizar tareas en el servidor");
}
else
{
    System.out.println("FIRMA KO. Error, firma inválida").
}
```

Once the "decodeMerchantParameters()" function has been called up, the value of any parameter susceptible for inclusion in the browsing control return can be obtained, as shown in section **Online response**. To obtain the value of a parameter we must call the "getParameter()" function of the library with

the parameter name as shown below to obtain the response code:

```
String codigoRespuesta = ApiMacSha256.getParame-
ter("Ds_Response");
```

IMPORTANT NOTE: It is important to validate all the parameters sent in the communication. To update the order status online this communication must NOT be used, but rather the online notification described in the other sections, as the browsing return depends on the client actions in the browser.

7.4.1.7.1.3 .NET library

Below are the steps to be followed by the merchant to use the .NET library provided by Redsys:

1. Import the library as shown below:

```
using RedsysAPIPrj;
```

2. Define an object of the chief library class, as shown below:

```
RedsysAPI r = new RedsysAPI();
```

3. Capture the browsing control return parameters:

```
String version = Request.QueryString["Ds_SignatureVer-
sion"];
String parms = Request.QueryString["Ds_MerchantParam-
eters"];
String signatureRecibida = Request.QueryString["Ds_Sig-
nature"];
```

IMPORTANT NOTE: It is important to validate all the parameters sent in the communication. To update the order status online this communication must NOT be used, but rather the online notification described in the other sections, as the browsing return depends on the client actions in the browser.

4. Validate the **Ds_Signature** parameter To validate this parameter, it is necessary to calculate the signature and compare it with the **Ds_Signature** parameter captured. To do so, call the function of the "createMerchantSignatureNotif f()" library using the merchant code provided and the **Ds_MerchantParameters** parameter captured as shown below:

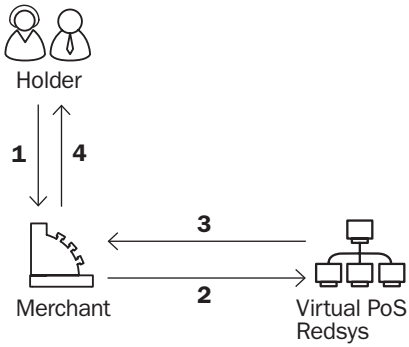
```
String clave = "sq7HjrUOBfkmC576ILgskD5srU870gJ7";
String signatureCalculada = r.createMerchantSignature-No-
tif(clave, data);
```

Upon completion, it is now possible to check whether the signature sent matches the value of the signature calculated, as shown below:

```
signatureReceived = r.GetParameter(XML, "<Signature>",
"</Signature>");
if (signatureCalculate = signatureReceived)
{
res = "FIRMA OK";
}
else
{
res = "FIRMA KO";
}
```

7.4.2 'WebService' Gateway:

The layout below shows the general flow of an operation performed with the Virtual PoS Web Service.



1. The holder selects the products at the merchant.
2. The merchant sends the payment data to the Virtual PoS.
3. Once payment is complete, the Virtual PoS informs the merchant of the result.
4. The merchant returns the payment result information to the holder.

7.4.2.1 Sending the request to the Virtual PoS

As shown in step 2 above, the merchant must send the payment request data via the Web Service to the Virtual PoS using UTF-8 encoding. The Web Service has several methods published on which operate the Virtual PoS. The "trataPeticion" method allows for operations via the Web Service, for which an XML must be built that includes the payment request data. The exact description of this XML request is presented via the WSDL file in Annex 5 (Web Service for payment request - WSDL) of the section Annexes in this document.

This payment request must be sent to the following URLs depending on whether a trial or real operation is to be performed.

Once the request is sent, the Virtual PoS will interpret and perform the relevant checks so as to process the operation as shown in step 3 above. Depending on the result of the operation a response XML document is built with the result of same with UTF-8 encoding.

7.4.2.2 Web Service payment request message

For the merchant to make the request via the Banco Sabadell WebService, it is necessary to exchange a series of data both in the request messages and the response messages.

The message structure will always be the same, with the **<REQUEST>** element as the root It must always contain three elements relating to:

- **Payment request data.** Element identified by the label <DATOSENTRADA>.
- **Version of the signature algorithm**. Element identified by the label <DS_SIGNATURE-VERSION>.
- Signature of the payment request data. Element identified by the label <DS_ SIGNATURE>.

Below is an example of the payment request message:

```
<REQUEST>
<DATOSENTRADA>
<DS_MERCHANT_AMOUNT>145</DS_MERCHANT_
AMOUNT>
<DS_MERCHANT_ORDER>151029142229</DS_MER-
CHANT_ORDER>
<DS_MERCHANT_MERCHANTCODE>327234688</
DS_MERCHANT_MERCHANTCODE>
<DS_MERCHANT_CURRENCY>978</DS_MERCHANT_
CURRENCY>
<DS_MERCHANT_PAN>4548812049400004</
DS_MERCHANT_PAN>
<DS_MERCHANT_EXPIRYDATE>1512</DS_MERCHANT_
EXPIRYDATE>
<DS_MERCHANT_CVV2>285</DS_MERCHANT_CVV2>
<DS_MERCHANT_TRANSACTIONTYPE>A</DS_MER-
CHANT_TRANSACTIONTYPE>
<DS_MERCHANT_TERMINAL>2</DS_MERCHANT_TER-
MINAL>
</DATOSENTRADA>
   <DS_SIGNATUREVERSION>HMAC_SHA256_V1</
DS_SIGNATUREVERSION>
   <DS_SIGNATURE>2YW19YQ8rb/0LLav79Y5L24Y-
w045KxN5hme27605WxY=</DS_SIGNATURE>
</REQUEST>
```

To facilitate the integration of the merchant, below is a detailed explanation of how to assemble the payment request message

7.4.2.3 Assembling the request data chain

A chain with all the data of the request must be generated in XML format to obtain as the result the <DATOSENTRADA> element.

- **Payment requests** (card data sent) Annex 1 (Payment requests) in the Annexes section presents the necessary parameters for this type of request, including an example.
- **Confirmation/Refund requests** Annex 3 (Confirmation/Refund requests) in the Annexes section presents the necessary

parameters for this type of request, including an example.

For merchants using special operations such as "Payment by Reference" (1-Clic [payment), they must include the specific fields of this type of operation in the <DATOSENTRADA> element.

7.4.2.4 Identifying the version of the signature algorithm to be used

It is necessary to identify in the request the specific version of algorithm being used for the signature. The value **HMAC_SHA256_V1** is currently used to identify the version of all requests so this will be the value of the **DS_SIGNATUREVERSION** parameter as can be seen in the lessage example shows at the start of section 3.

7.4.2.5 Identifying the code to be used for signing

To calculate the signature it is necessary to use a specific code for each terminal. The merchant code to be used is that received from Banco Sabadell in an SMS.

IMPORTANT NOTE: This code must be stored in the merchant server as secure as possible to avoid any fraudulent use of same. The merchant is responsible for properly safeguarding and maintaining the code secret.

7.4.2.6 Signing the request

Once the element with the payment request data has been generated (<DATOSENTRADA>) and the specific code of the terminal, the signature must be calculated by applying the following steps:

1. A specific code is generated per operation. To obtain the code to be used in an operation 3DES encryption must be made between the merchant code, which must be first decoded in BASE 64, and the value of the operation order number (Ds_Merchant_Order).

2. The HMAC SHA256 of the <DATOSENTRADA> element is calculated

3. The result obtained is encoded in BASE 64, and the result will be the value of the <DS_SIGNATURE> element, as shown in the example of the form at the beginning of section 3.

7.4.2.7 Use of the help libraries

The previous sections described the method for accessing the SIS using a connection via the Web Service and the signature system based on HMAC SHA256. This section explains how the libraries available in PHP and JAVA are used to facilitate developments and generate signatures.

7.4.2.7.1 PHP library

Below are the steps to be followed by the merchant to use the PHP library provided by Banco Sabadell:

1. Import the chief library file as shown below:

```
include './apiRedsysWs.php';
```

   The merchant must decide whether to perform the import using the "include" or "required" function, in accordance with the developments made.

2. Define an object of the chief library class, as shown below:

```
$smiObj = new RedsysAPIWs;
```

3. Calculate the <DS_SIGNATURE> element.

To calculate this parameter, call up the library function “createMerchantSignatureHostTo- Host()” with the merchant code provided and the element of the payment request data (<DATOSENTRADA>), as shown below:

```
$datoEntrada='<DATOSENTRADA><DS_MERCHANT_AMOUNT>'.$importe.'</DS_MERCHANT_AMOUNT><DS_MERCHANT_ORDER>'
$clave = 'sq7HjrU0BfKmC576ILgskD5srU870gJ7';
$signature = $miObj->createMerchantSignatureHostTo-Host($clave, $datoEntrada);
```

Once the value of the <DS_SIGNATURE> has been obtained, it is possible to complete the payment request message and perform the Web Service call.

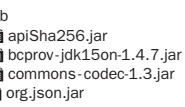
7.4.2.7.2 Java library

Below are the steps to be followed by the merchant to use the JAVA library provided by Banco Sabadell:

1. Import the library as shown below:

```
<%@page import="sis.redsys.api.ApiWsMacSha256"%>
```

The merchant must include in the prokect construction the libraries (JARs) provided:



2. Define an object of the chief library class, as shown below:

```
ApiWsMacSha256 apiWsMacSha256 = new ApiWsMacSha256();
```

3. Calculate the **<DS_SIGNATURE>** element. To calculate this parameter, call up the library function “createMerchantSignatureHostTo- Host()” with the merchant code provided and the element of the payment request data (**<DATOSENTRADA>**), as shown below:

```
String datosEntrada = "<DATOSENTRADA><DS_MERCHANT_AMOUNT>200</DS_MERCHANT_AMOUNT>..."
String clave = "sq7HjrU0BfKmC576ILgskD5srU870gJ7";
String firma = apiWsMacSha256.createMerchantSignatureHostToHost(clave, datosEntrada);
```

Once the value of the **<DS_SIGNATURE>** element has been obtained, it is possible to complete the payment request message and perform the Web Service call.

7.4.2.7.3 .NET library

Below are the steps to be followed by the merchant to use the .NET library provided by Banco Sabadell:

1. Import the library as shown below:

```
Using RedsysAPIPrj;
```

2. Create an object of the Redsys Web Service object. To do this it is necessary to add a new web reference with the file SerClsWSEntrada.wsdl.

```
WebRedsysApi.WebRedsysWs.SerClsWSEntradaService s = new WebRedsysAPI.WebRedsysWs.SerClsWSEntradaService();
```

NOTE: In the location attribute of the label <wsdlsoap:address> of the SerClsWSEntrada.wsdl file, indicate whether it is a real or test environment:

```
https://sis-t.redsys.es:25443/sis/services/SerClsWSEntrada (Pruebas)
https://sis.redsys.es/sis/services/SerClsWSEntrada (Real)
```

3. Define an object of the chief library class, as shown below:

```
RedsysAPIWs r = new RedsysAPIWs();
```

When taking this step the dictionary code/value m_keyvalues and cryp of the Cryptogra attributes are initialised (Ancillary class for performing the necessary cryptographic operations)

4. Generate DATOSENTRADA parameters (Payment Request type sending card data) using the function:

```
string datoEntrada = r.GenerateDatoEntradaXML(importe, merchantCode, moneda, numTarjeta, cvv2, transactionType, terminal, expiryDate);
```

5. Calculate the **<DS_SIGNATURE>** element. To calculate this parameter, call up the library function "createMerchantSignatureHostTo- Host()" with the code obtained from the administration module and the element of the payment request data (**<DATOSENTRADA>**), as shown below:

```
string signature = r.createMerchantSignatureHostToHost(clave, datoEntrada);
```

Once the value of the <**DS_SIGNATURE**> element has been obtained, it is possible to complete the payment request message and perform the Host to Host Service call.

The final XML string of the payment request is generated using DATOSENTRADA, DS_SIGNATUREVERSION and DS_SIGNATURE calculated in point 5.

```
string requestXML = r.GenerateRequestXML(datoEntrada, signature);
```

After calling up the trataPeticion method of the Redsys Web Service sending the final XML string as a parameter calculated with the GenerateRequestXM methodL.

```
string result = s.trataPeticion(requestXML);
```

7.4.2.8 Response of the Web Service request

Web Service. This message is generated in XML format and an example is shown below:

```
<RETORNOXML>
  <CODIGO>0</CODIGO>
        <OPERACION>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>
        <Ds_Order>151029142229</Ds_Order>
        <Ds_Signature>MRvyhuDEpg4BmzfTdgHKrl-
        5qQ9U5UD2Qe8eDadIZtyE=</Ds_Signature>
        <Ds_MerchantCode>327234688</Ds_Merchant-
        Code>
        <Ds_Terminal>2</Ds_Terminal>
        <Ds_Response>0000</Ds_Response>
        <Ds_AuthorisationCode>185714</Ds_Authorisation-
        Code>
        <Ds_TransactionType>A</Ds_TransactionType>
        <Ds_SecurePayment>0</Ds_SecurePayment>
        <Ds_Language>1</Ds_Language>
        <Ds_MerchantData></Ds_MerchantData>
        <Ds_Card_Country>724</Ds_Card_Country>
  </OPERACION>
</RETORNOXML>
```

As we can see in the example above the response comprises two chief elements:

- Code (**<CODIGO>**): Indicates whether or not the operation was correct (does not indicate if it was authorised, only if processed). An 0 indicates that the operation was correct. If different from 0, it will have a code. (See error codes in section 5 of this Guide)
- Data of the operation (**<OPERACION>**): This gathers all the necessary information about the operation performed. Using this element we determine whether or not the operation has been authorised.

NOTE: The list of parameters forming part of the response is described in the Annexes.

7.4.2.8.1 Signing the response message

Once the response message and specific code of the terminal have been obtained, provide the operation is authorised, it is necessary to check the response signature by applying the following steps:

1. A specific code is generated per operation. To obtain the code to be used in an operation, 3DES encryption must be made between the merchant code, which must be first decoded in BASE 64, and the value of the operation order number (DS_ORDER).
2. The HMAC SHA256 of the chain formed by the concatenation of the value of the following fields is calculated:

```
Cadena = Ds_Amount + Ds_Order + Ds_Merchant-
```

```
Code + Ds_Currency + Ds_Response + Ds_CardNumber + Ds_TransactionType + Ds_SecurePayment
```

If we take as an example the response presented at the beginning of this section, the resulting chain would be:

```
Cadena = 1451444912789999008881978000000
```

If the merchant has configured the sending of the asterisked card in the response, it is necessary to calculate the HMAC SHA256 of the chain formed by the concatenation of the value of the following fields:

```
Cadena = Ds_Amount + Ds_Order + Ds_MerchantCode + Ds_Currency + Ds_Response + Ds_CardNumber + Ds_TransactionType + Ds_SecurePayment
```

If we take as an example the response presented at the beginning of this section, the resulting chain would be:

```
Cadena = 145144982154599900888197800004548 81******000400
```

3. The result obtained is encoded in BASE 64, and the result of the encoding must be the same as the value of the **Ds_Signature** obtained in the response.

7.4.2.8.2 Use of the help libraries

This section explains how the libraries available in PHP and JAVA are used to facilitate developments and generate signatures.

7.4.2.8.2.1 PHP library

Below are the steps to be followed by the merchant to use the PHP library provided by Banco Sabadell:

1. Import the chief library file as shown below:

```
Include './apiRedsysWs.php';
```

The merchant must decide whether to perform the import using the "include" or "required" function, in accordance with the developments made.

2. Define an object of the chief library class, as shown below:

```
$miObj = new RedsysAPIWs;
```

3. Calculate the **Ds_Signature** parameter To calculate this parameter call up the library function"createSignatureResponseHostToHost( )" using the merchant code provided, the chain to be signed (concatenation of fields described in point 2 of section 4.1 of this document) and the order number.

```
$cadenaConcatenada = "145151029142229327234 6889780000A0";
$numPedido = "151029142229";
$clave = 'sq7HjrUOBfKmC576ILgskD5srU870gJ7';
$signature = $miObj->createMerchantSignatureResponseHostToHost($clave, $cadenaConcatenada, $numPedido);
```

The result obtained must be the same as the value of the **Ds_Signature** parameter obtained in the response.

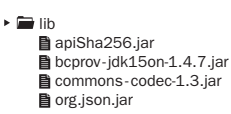
7.4.2.8.2.2 Java library

Below are the steps to be followed by the merchant to use the JAVA library provided by Banco Sabadell:

1. Import the library as shown below:

```
<%@page import="sis.redsys.api.ApiWsMacSha256"%>
```

The merchant must include in the prokect construction the libraries (JARs) provided:



2. Calculate the **Ds_Signature** parameter To calculate this parameter call up the library function"createSignatureResponseHostToHost( )" using the merchant

code provided, the chain to be signed (concatenation of fields described in point 2 of section 4.1 of this document) and the order number.

```
String cadenaConcatenada = "14515102914222932
72346889780000A0";
String numPedido = "145151029142229327234688
89780000A0";
String clave = "sq7HjrUOBfKmC576ILgskD5s-
rU870gJ7";
String signature = apiMacSha256.createMerchantSig-
natureResponseHostToHost(clave, cadenaConcatena-
da, numPedido);
```

The result obtained must be the same as the value of the **Ds_Signature** parameter obtained in the response.

7.4.2.8.2.3 .NET library

Below are the steps to be followed by the merchant to use the .NET library provided by Redsys:

1. Convert the XML response chain to the dictionary attribute m_keyvalues of the RedsysAPIWs code:

```
r.XMLToDiccionary(result);
```

2. Calculate the **Ds_Signature** parameter To calculate this parameter call up the library function "createSignatureResponseHostToHost()" using the administration module code provided, the chain to be signed (concatenation of fields described in point 2 of section 5.1 of this document) and the order number.

```
string cadena = r.GenerateCadena(result);
string numOrder = r.GetDictionary("Ds_Order");
string signatureCalculate = r.createSignatureRespon-
seHostToHost(kc, cadena, numOrder);
```

The result obtained must be the same as the value of the **Ds_Signature** parameter obtained in the response.

7.4.3 General recommendations for calculating the signature

Once the signature has been generated the data of the requisition should never be modified as the Virtual POS uses them to validate it. If the data sent to the Virtual POS are not exactly the same as those used to generate the signature, an error will occur and the purchase cannot continue.

The **Amount** will be multiplied by 100, without decimals or zeros to the left, except in the case of yen which have no decimals.

The **order number** will be different in each transaction and the first 4 positions must be numerical.

Check that the **code** being used for the signature is that assigned to the merchant and check the environment (test or real), in which the purchase requisition is being made.

IMPORTANT NOTE:

- The secret code must never be disclosed to third parties, nor appear in the source code of the merchant's website nor be accessible within the website's file structure.
- The calculation of the Hash SHA-256 algorithm must be implemented on the private section of the merchant's Internet server.
- If the merchant resides on an unrelated server under a hosting arrangement or similar, contact the provider to ascertain how to implement the cryptographic algorithm.

7.5 Online response from the Virtual POS to the merchant

This option is available for those merchants require immediate verification of the transactions carried out via the Virtual POS for their processes.

There are various response systems which can coexist simultaneously. They are as follows:

1. Direct query of transactions via the Internet by accessing the **Virtual POS administration module**.
2. Implementation of an **Online response** solution.

   At the same time the cardholder receives the response to the card payment requisition, the merchant website receives a message with the same information.

3. **Reception of data via XML..**

   This type is available solely for the "Gateway-Web- Service." Type.

Below is further information on the "Online response" which is the most common system.

7.5.1. Online response

There are two online response reception modes which can be combined, using both at the same time or one of them as a backup if the other fails:

Email notification to merchant:
The response to the payment authorisation will be received at the email address the merchant indicated when requesting registration of the Virtual POS.

HTTP notification:

The response to the payment authorisation will be received at the URL address (Ds_Merchant_MerchantURL) indicated on the payment form. This option requires some simple program development on the merchant's website. **This option is recommended as it guarantees immediate response**.

To implement the online response via HTTP Notification the payment requisition form must provide an URL for reception of the responses (Ds_Merchant_MerchantURL field). This URL will be a CGI, Servlet or similar, developed in the language considered suitable for the merchant server to interpret the response sent by the Virtual POS. The URL will not be loaded in the browser and will therefore not be visible to the user. It can receive and collect data of the online response and enter them in the merchant's database.

The protocol used in response via URL can be http or https, the format of this message is an HTML form, sent via POST, and whose fields are as follows:

DATUM	NAME OF FIELD	COMMENTS
Signature version:	Ds_SignatureVersion	Constant indicating the signature version being used.
Details of the operation	Ds_MerchantParameters	Chain in JSON format with all the parameters of the request encoded in Base 64.
Signature	Ds_Signature	Result of the HMAC SHA256 of the encoded JSON chain in Base 64 sent in the previous parameter.

Para acceder a los datos de la operación, los datos deberán ser desencriptados. Esta desencriptación se realiza en el momento en el que se genera la firma de notificación, tal y como se indica en los ejemplos.

DATUM	NAME OF FIELD	LENGHT/TYPE	COMMENTS
Date	Ds_Date	dd/mm/yyyy	Transaction date.
Time	Ds_Hour	HH:mm	Transaction time.
Amount	Ds_Amount	12 / No.	Same value as in requisition.
Currency	Ds_Currency	4 / No.	Same value as in requisition. 4 is considered the maximum length.
Order number	Ds_Order	12 / A-N.	Same value as in requisition.
Merchant Code/FUC Code	Ds_ MerchantCode	9 / No.	Same value as in requisition.
Terminal	Ds_Terminal	3 / No.	Same value as in requisition. 3 is considered the maximum length.
Response code	Ds_Response	4 / No.	See following table (Possible values of Ds_Response).
Merchant data	Ds_ MerchantData	1024 / A-N	Optional information sent by merchant in payment requisition form.

DATUM	NAME OF FIELD	LENGHT/TYPE	COMMENTS
Secure payment	Ds_SecurePayment	1 / Núm.	0 – If payment is NOT secure 1 – If payment is secure.
Type of transaction	Ds_TransactionType	1 / A-N	Type of operation sent in the payment form.
Holder's country	Ds_Card_Country	3/No.	Country of card issuance See ANNEX 1 with list of countries.
Authorisation code	Ds_Authorisation Code	6/ A-N	**Optional**: Alphanumerical code of authorisation assigned to approval by authorising institution.
Holder's language	Ds_Consumer Language	3 / No.	**Optional**: The value 0 will indicate that customer language has not been determined. (Optional) 3 is considered the maximum length.
Card type	Ds_Card_Type	1 / A-N	**Optional**: Possible values: C - Credit Card D - Debit card
Card number	Ds_Card_Number	15-19/A-N	**Optional**: The value of this variable will be the card number with an asterisk. By default, this variable is not activated.
Identifier	Ds_Merchant_ Identifier	40/A-N	Value of the field is Required for first payment transaction. This variable will only be sent if reference payment operations are activated
Expiry date	Ds_ExpiryDate	4 / N	Expiry date of the card. This variable will only be sent if reference payment operations are activated

(In the Ds_Currency, Ds_Terminal and Ds_ConsumerLanguage fields the length is considered the maximum so it is unnecessary to fill in with zeros to the left. The signature will be generated with the fields exactly as sent.)

The connection used to communicate the online confirmation between the Virtual PoS and the merchant can be TLS when using a security certificate (https).

The default Virtual POS can communicate with ports 80, 443, 8080 and 8081 of the merchant. Other ports must be checked with the Banco Sabadell technical service.

Once the merchant receives the form, the values of the Response code field (Ds_Response) indicate whether the operation is approved or rejected and if so, the reason for rejection. Annex III includes the table of response codes.

The Virtual POS sends the online notifications for purchase operations authorised and rejected by the card issuing entity as well as in those situations in which the purchase process was interrupted due to one of the following errors:

SIS0051 -> Order repeated. A notification with code 913 is sent.

SIS0078 -> Method of payment not available for the card. A notification with code 118 is sent.

SIS0093 -> Invalid card. A notification with code 180 is sent.

SIS0094 -> Error in calling MPI not controlled. A notification with code 184 is sent.

SIS0218 -> The merchant does not allow pre-authorisation by XML input.

SIS0256 -> Merchant cannot perform pre-authorisations.

7.6 Payment of subscriptions and payments express

In order to increase the conversion rate and facilitate as far as possible the process of purchase, Banco Sabadell Virtual POS incorporates an innovative feature that allows payment of subscriptions and payments express through an identifier equivalent to the number of card.

This method allows more easily manage purchases by regular customers, because they do not need to enter the card data in each transaction. The buyer only has to fill the card details in the first purchase. The merchant will receive, together with the payment response, an identifier for use in subsequent purchases. In addition, the merchant will be informed of the expiry date of the card and optionally the number of the card, properly masked, ie with certain digits replaced by asterisks.

The card details are stored on the servers of Banco Sabadell and thus avoid the merchant having to fulfill the PCI -DSS security requirements. (See Chapter 11)

- **Operating process for the first payment:**

  The merchant requests Virtual POS for payment. Together with the necessary payment details, a new parameter is sent to request generation of an identifier associated with the card details. This request may be made via any of the current Virtual POS gateways (realizarPago, entradaXMLentidad, operaciones or WebService).

  If the merchant has not sent the card, the Virtual POS will request same from the holder together with the expiry date and CVV2.

  The Virtual POS processes the payment request and stores the card details associated with an internally generated ID. The identifier will only be generated if payment is authorised.

  The Virtual POS returns the identifier and expiry date together with the payment

response so the merchant can use it subsequently. Optionally, the Virtual POS can send together with the payment message, the number of the card properly masked.

Depending on the type of connection used by the merchant, the reference will be returned via the following channels

i. **For the 'realizarPago' input**: The reference and expiry date will be returned in the On-Line notification and OK URL.

ii. **For the 'webService' input:** the reference and expiry date will be returned in the response of authorised transactions.

- **Operating process for the subsequent payments:**

  Once the merchant has an identifier, it can use it in subsequent payments instead of sending the card and expiry date. The operating process would be as follows:

  – New payment: The merchant requests payment from Virtual POS and therefore sends the identifier than Banco Sabadell sent in the first payment.

  – The expres payment/subscription payment operation is valid for any type of transaction (Ds_Merchant_Transaction Type).

  – The merchant may opt to indicate whether or not it wishes to display additional screens (DCC, Splitting and Authentication).

  – The merchant can use any gateway to the Virtual POS of those available (realizarPago or WebService).

  – The Virtual POS validates the identifier associated with the merchant and recovers the card details.

  – Once the card details have been located, the Virtual POS proceeds to make payment. If it was chosen not to display screens, payment will be made without showing the DCC or splitting screens and without using a secure payment method. The expiry date will only be included in the reply if the merchant is configured for this.

For those cases in which the merchant, when applying for a payment to Virtual POS, has not requested a creation of an identifier, or were using the previous payment method of Banco Sabadell, called 'Card on File', will be possible to create identifier afterwards. To do Banco Sabadell has a batch process called 'GenerarReferencias', through which you can filter the transactions for which you want to create identifiers.

Annex VI of this manual provides specific examples using the Payment of Subscriptions / Express Payments functionality, for each of modes of processing transactions through the Banco Sabadell Virtual POS.

Restricions

A merchant using this operation must bear in mind the following restrictions:

i. The identifier will also be associated with the number of the merchant making the request. If the merchant wants this identifier to be used by other merchants, they must be previously configured to form a group. In order to form groups, you must ask your Banco Sabadell account manager.

ii. The card details will be stored until expiry of its validity date.

iii. The validity of the identifier will be limited to the expiry date of the card and will always be returned in the response when a new identifier is requested. In other cases it will only be returned in

the response to merchants that are configured accordingly.

iv. It is only possible to not display screens when a valid reference is used. When generation of a new identifier is requested and in any other case, the merchant cannot request that screens not be shown.

The other parameters necessary for payment do not differ from those of an ordinary payment.

- **Ds_Merchant_Identifier**

  This parameter will be used to handle the reference associated with the card details. It is an alphanumeric field with a maximum of 40 positions whose value is generated by the Virtual POS.

  1st Request: In the first request for the merchant to seek generation of a new identifier the value "REQUIRED" must be sent. The Virtual POS will return the generated identifier associated with the card in a parameter with the same name. The Virtual POS will always return the expiry date which will be in the parameters **Ds_ExpiryDate**. As we indicated above, both parameters will be return in the online Notification, URL OK or response to WebService depending on the connection used by the merchant.

  The Ds_Merchant_Identifier parameters must be included in the Hash signature calculation chain (see section 7.6.4 of this manual). It **must be concatenated at the end of the data chain** and before the value of the code or Ds_Merchant_Group parameter if any.

  2nd and successive requests: The merchant must send the reference in the Ds_Merchant_Identifier parameter and not provide the card details. The expiry date will only be included in the reply if the merchant is configured for this.

  The Ds_Merchant_Identifier parameters **must be included in the Hash signature calculation chain** (see section 7.6.4 of this manual). It must be concatenated at the end of the data chain and before the value of the code or Ds_Merchant_Group parameter if any or the Ds_Merchant_DirectPayment parameter if any and the Ds_MerchantGroup parameter does not exist.

- **Ds_Merchant_Group**

  This parameter associates a reference to a set of merchants. It is an optional numerical parameter with a maximum of 9 positions. If this parameter is sued the reference will be associated with the group code instead of the merchant code.

  The group of merchants must first be defined in the Virtual PoS. To create groups it is necessary request this from your regular Banco Sabadell agent.

  If a reference is associated with a group of merchants, each of the merchants can then individually use it.

  This parameter **must be included in the Hash signature calculation chain** (see section 7.6.4 of this manual). It must be concatenated just after the Ds_Merchant_Identifier parameter and before the value of the code or the Ds_Merchant_DirectPayment parameter if any.

- **Ds_Merchant_DirectPayment**

  This parameter acts as a flag to indicate if additional screens must be shown (DCC, Splitting and Authentication). It is an optional parameter that can only

have the values “true” or “false”. Itf used with the value “true”, no additional screens will be displayed (DCC, Splitting and Authentication) during payment and it must be used in conjunction with the Ds_Merchant_Identifier parameter containing a valid reference. If it is not used or is used with the value “false”, payment will be made as usual and all the additional screens will be displayed (DCC, Splitting and Authentication) which are required depending on the merchant configuration.

This parameter **must be included in the Hash signature calculation chain** (see section 7.6.4 of this manual). It must be concatenated just after the Ds_Merchant_Group parameter (if any) and before the value of the code.

Migration of identifiers

(Only for those merchants that are already using the existing Banco Sabadell payment method called ‘Card on File’)

A merchant can continue to use the existing Card on File operation until that time or start to use Payment by Reference.

In some cases, the merchant will want to use the new operation for previous transactions. To do this a identifier migration process has been developed, from Card on File operations to the new Payment of Subscriptions / Payments Express.

The migration of identifiers will be made by means of an express request to your Banco Sabadell account manager. Once the request has been processed, the merchant will have a file with the following data by transaction:

- Merchant ID
- Terminal No.
- Transaction Date
- Original operation order code
- Identifier generated and recorded for the card of the original transaction

With this file the merchant can update its systems to use identifiers.

Example of file with identifiers

Merchant;Terminal;Order;Date;Identifier

999008881;1;130211123726;2013-02-11-12.37.27.381; 7490da446dee0a...25b6b-d52e086c3181

999008881;1;130211123739;2013-02-11 12.37.40.429;d5ac083cb97d183...548f16 8c32c7bb5ab7d

7.7 DCC operational settings

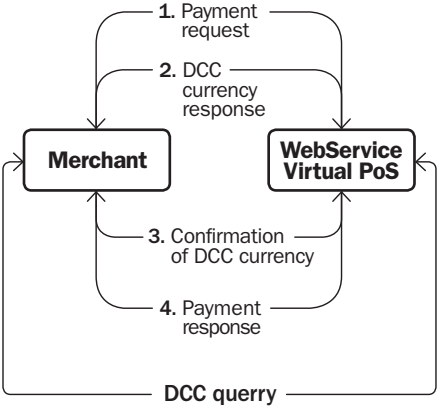
The Virtual POS of Banco Sabadell allows to both holders of Visa or MasterCard cards issued in a currency other than the euro, to pay for purchases done, in the original currency of the card. See details in section 5.4 of this manual.

The following are additional technical features that are required for configuring the operating DCC in the merchant’s Virtual POS. However, is it only required to set the Virtual POS, if merchant uses the ‘WebService’ Gateway, as described in paragraph 7.4 of this manual.

In the case of using the ‘Operaciones’ gateway, you cannot configure the operating DCC since the merchant at any time gives the Bank the browser session and therefore we cannot display neither the selection screens currency nor the receipt of DCC purchase.

For the entry no configuration is required. The DCC currency selection screens and the DCC sales receipt will automatically appear if the Virtual PoS detects a transaction

using Visa or MasterCard cards issued in a currency other than the euro.



NOTE: As show in the chart, the DCC operations are based on sending two requests to the WebService of the Virtual PoS. To ensure proper operation of the system, the merchant must mtain the session between the first and second call to the WebService. Session maintenance will depend on the software used to make the call to the WebService. For example, if the Axis API is used it is sufficient to use the same "Stub" for the two requests and fixed the property "setMaintainSession(true)" before making the first call.

7.7.1 Access methods

The access method "trataPeticion": Allows operations via the WebService Virtual PoS. The same method is used for traditional payments and DCC operations and one or the other option will be used in accordance with the fields sent in the request XML.

The access method "consultaDCC": allows queries to the DCC associated with an amount and a currency prior to executing the transaction. It is purely for information purposes.

7.7.2 'Webservice' Gateway – DCC Operational settings

1. Initial payment request message

The initial request message (1. Payment request) has the same characteristics as described in section 7.4.1 of the present manual. In this type of message no special encoding is required to activate the DCC operations.

2. DCC response message

Described below are the necessary data and their characteristics, which will be received in the DCC response messages (2. DCC currencies response) of the Virtual PoS in XML format described above for DCC operations and which are an example for subsequent DCC confirmation.

DATA	LENGHT/TYPE	DESCRIPTION
Moneda	3 / N	**Obligatory.** Value of currency identifier.
litMoneda	- / A	**Obligatory.** Literal associated to currency.
litMonedaR	3 / R	**Obligatory.** Reduced literal associated with currency.

DATA	LENGHT/TYPE	DESCRIPTION
cambio	- / N	**Obligatory.** Currency exchange value.
importe	- / N	**Obligatory.** Amount in currency.
checked	true / false	**Obligatory.** Indicates checked currency.
margenDCC	- / N	**Obligatory.** DCC margin applied by the bank to the amount.
nombreEntidad	- / A	**Obligatory.** Name of the bank applying the DCC.
DS_MERCHANT_ SESION	- / AN	**Obligatory.** Session identifier to continue the transaction in DCC operations.

Type A: ASCII code 65 characters = A to 90 = Z and 97 = a to 122 = z.
Type N: ASCII code 30 characters = 0 to 39 = 9

Example of DCC response

```
<RETORNOXML>
   <CODIGO>0</CODIGO>
   <DCC>
         <moneda>826</moneda>
         <litMoneda>POUND STERLING</litMoneda>
         <litMonedaR>GBP</litMonedaR>
         <cambio>1.413788</cambio>
         <importe>1.03</importe>
         <checked>true</checked>
   </DCC>
   <DCC>
         <moneda>978</moneda>
         <litMoneda>Euros</litMoneda>
         <importe>1.45</importe>
   </DCC>
   <margenDCC>2.5</margenDCC>
   <nombreEntidad>SIN CAPTURA</nombreEntidad>
   <DS_MERCHANT_SESION>vXYlxTsfkVJ6ZL82vJ48Lvm</DS_
   MERCHANT_SESION>
</RETORNOXML>
```

3. DCC Confirmation Message

Described below are the necessary data and their characteristics, for sending a DCC confirmation request (3. DCC currency confirmation):

DATA	LENGHT/TYPE	DESCRIPTION
Sis_Divisa	16/A-N	**Obligatory.** Two values separated by #. The first is the currency identifier, the second the amount in that currency.
DS_MERCHANT_ SESION		**Obligatory.** Session identifier to continue the transaction in DCC operations.

Type N: ASCII code 30 characters = 0 to 39 = 9

Example of DCC currency confirmation message

```
<REQUEST>
   <DATOSENTRADA>
      <DS_MERCHANT_ORDER>0804620125</DS_MERCHANT_
      ORDER>
      <DS_MERCHANT_MERCHANTCODE>327234688</DS_MER-
      CHANT_MERCHANTCODE>
      <DS_MERCHANT_TERMINAL>2</DS_MERCHANT_TERMI-
      NAL>
      <SIS_DIVISA>826#1.03</SIS_DIVISA>
      <DS_MERCHANT_SESION>vXYlxTsfkVJ6ZL82vJ48Lvm</
      DS_MERCHANT_SESION>
   </DATOSENTRADA>
   <DS_SIGNATUREVERSION>HMAC_SHA256_V1</DS_SIGNA-
   TUREVERSION>
   <DS_SIGNATURE>iJ13pCEL09CmJ8hosYjyWWUF/KYdPb1vsSu-
   WGl3k1zg=</DS_SIGNATURE>
</REQUEST>
```

4. Response message

The response message (4.Payment response) possesses the same characteristics as described in section 7.4.2.8 of the present manual. In this type of message no special encoding is required to activate the DCC operations.

5. DCC query message

The DCC query message will be generated with the data previously described in an XML that will be sent to the consultaDCC method. This query is only for informational purposes.

Example of DCC response

```
<REQUEST>
   <DATOSENTRADA>
   <DS_MERCHANT_AMOUNT>1.06</DS_MERCHANT_AMOUNT>
   <DS_MERCHANT_ORDER>1444904795</DS_MERCHANT_OR-
   DER>
   <DS_MERCHANT_MERCHANTCODE>999008881</DS_MER-
   CHANT_MERCHANTCODE>
   <DS_MERCHANT_TERMINAL>6</DS_MERCHANT_TERMINAL>
   <DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CURREN-
   CY>
   </DATOSENTRADA>
   <DS_SIGNATUREVERSION>HMAC_SHA256_V1</DS_SIGNA-
   TUREVERSION>
   <DS_SIGNATURE>oVGAkwOQNYHqDN8+i2oBRKYn8aZR4s7L-
   JOcHpwnuCoU=</DS_SIGNATURE>
</REQUEST>
```

6. DCC query response message

The DCC query message will be generated with the data described above in an XML which will be sent to the consultaDCC method.

Example of DCC query response message

```
<RETORNOXML>
    <CODIGO>0</CODIGO>
    <DCC>
    <moneda>978</moneda>
    <importe>0.01</importe>
    </DCC>
    <margenDCC>0.03</margenDCC>
    <nombreEntidad>SIN CAPTURA</nombreEntidad>
</RETORNOXML>
```

7.7.3 Merchant signature

DCC confirmation requests

If it is necessary to make a second request to the WebService Virtual PoS, as DCC is accepted in the operation, the signature will be calculated in the same manner as the first request.

Dcc query requests

The electronic signature of the merchant must be calculated in the same manner as a Webservice request:

7.8 Test environment

The test environment makes it possible to conduct the necessary tests to check for proper system operation before using the merchant's Virtual POS in a real environment. This environment is identical to the real environment but without the payments having accounting validity.

The test environment codes provided below are common to other customers of Banco Sabadell. If you wish to have test codes exclusively for your merchant, please contact the Technical Support Service to request installation of the Banco Sabadell Virtual POS.

The test environment parameters are:

1. URLs for sending the payment orders:

   Input "realizarpago (HTML)":
   https://sis-t.redsys.es:25443/sis/realizarPago

   Input "WebService (XML)":
   https://sis-t.redsys.es:25443/sis/services/SerClsWSEntrada

2. Merchant number (Ds_Merchant_MerchantCode): 327234688

3. Secret code (Ds_Merchant_MerchantSignature): sq7HjrUOBfKmC576ILgskD5srU870gJ7

4. Terminals (Ds_Merchant_Terminal):

   - 001 - For payments in EUROS (Ds_MerchantCurrency = 978) of merchants under protocol 3D Secure (Secure eCommerce –VERIFIED BY Visa y MasterCard SecureCode–)
   - 002 - For payments in EUROS (Ds_MerchantCurrency = 978) of merchants under Non-3D Secure protocol (payments considered INSECURE)

5. Card accepted:

   - 4548 8120 4940 0004, expiry date 12/20, CVV2 Code: 533
   - For 3D Secure payments in which purchaser authentication is required, the Personal Identification Code (PIC) is: 123456

For CES payments requiring purchaser authentication, the personal identification code (PIC) is 123.

6. URL Administration Module: https://sis-t.redsys.es:25443/canales/bsabadell

7. Access to administration module:

   » For terminal 001 (CES):
   User: 327234688-001
   Password: 123456a

   » For terminal 002 (NO CES):
   User: 327234688-002
   Password: 123456a

7.9 Technical support service for installation

To offer all the necessary support during the registration and installation process of the Banco Sabadell Virtual POS, we place at your disposal a specialised support service:

Service timetable:
Monday to Sunday, from 8 am to 10 pm
Telephone no: 902 365 650 (opt. 2)
Email:
tpvvirtual@bancsabadell.com

Vip technical support

In addition, for those clients who due to technical or commercial characteristics require urgent technical and/or preferential assistance, we place at their disposal a VIP technical support service.

Consult your e-commerce delegate about the conditions for obtaining this service.

Email:
ServicioTPVVirtualPlus@bancsabadell.com

Only in cases of **communication-related incidents, system instability and similar, please call 902 198 747 24**, hours a day every day of the year (support service provided by RedSys).

8.
SOAP-XML query of virtual POS transactions

SOAP is a protocol standard based on XML which enables communications with Web services. SOAP provides a simple, uniform system for sending XML messages to another application.

Below we describe the steps to following to use the SOAP query web service on transactions with a view to making queries on operations carried out by the merchant. Regardless of whether they have been answered. Therefore, the online query service offers the possibility of obtaining information on all operations which have commenced.

There are various types of **queries: by transaction and by monitor**, mass or otherwise.

Query by transaction offer information on a certain type of operation (e.g. Authorisation) of an order.

Query by monitor offers information on all types of operation (e.g. Authorisation and Refund) associated with an order number.

These operations can be **simple** (for a single date) or **mass** (between a range of dates).

The possible values of a query by transaction are:

<Ds_TransactionType> = **0** (Normal payment operation)
<Ds_TransactionType> = **1** (Unconfirmed pre-authorisation)
<Ds_TransactionType> = **4** (Payment by reference)
<Ds_TransactionType> = **7** (Pre-authentication)
<Ds_TransactionType> = **A** (Non-secure payment without authentication)

The specification of the service input and output messages is described in Section 8.2 of the present manual.

The access service to card management is implemented using SOAP-XML technology. Simple Object Access Protocol (SOAP). To use this service it is necessary to use this technology.

This service easily enables the terminal to make an operation requisition with the cards. The service requester will make a requisition to the service provider who will return the result of same.

It is necessary to implement a SOAP client which will call the method responsible for initiating the transaction (see example in Section 8.4 of the present manual).

The SOAP client must send an XML to the Web Service Access Service which will contain the data of the operation to be performed. This will send a response XML. These XML must comply with XML-SCHEMA under the terms described in the present manual.

8.1 Calculating the signature

Due to the confidential nature of the data sent in the messages, it is necessary to protect these data such that no outside party can modify them and confuse one of the two parties involved in the system.

In addition to the inherent security provided by SOAP when sending the messages, it is necessary to add a signature to the messages which identifies the two parties to the transaction. To calculate the signature, a secret code is used which is shared by the merchant and Banco Sabadell.

The data included in the signature are a chain with the message sent (the data between the labels <version...> </version>) and the secret code shared between Banco Sabadell and the merchant.

For example given the following request data:

```
<Version Ds_Version="0.0">
<Message>
<Monitor>
<Ds_MerchantCode>327234688</Ds_MerchantCode>
<Ds_Terminal>1</Ds_Terminal>
<Ds_Order>91031000014</Ds_Order>
</Monitor>
</Message>
</Version>
```

Will generate the following XML:

```
<Messages>
<Version Ds_Version="0.0">
<Message>
<Monitor>
<Ds_MerchantCode>327234688</Ds_MerchantCode>
<Ds_Terminal>1</Ds_Terminal>
<Ds_Order>91031000014</Ds_Order>
</Monitor>
</Message>
</Version>
<Signature>KYMRHr7g9at+t2xTx7Mem3pW52rkCckMpd5x13Tr-
D5I=</Signature>
<SignatureVersion>HMAC_SHA256_V1</SignatureVersion>
</Messages>
```

This gives the value 571b2d002c878ddb241fc-542c7b6d46262bbb7cb.

To calculate the response signature the following data are used, as with the input signature.

8.2 SOAP queries – Specification of incoming and outgoing messages

This section describes the details of the incoming and outgoing messages of the Web service for transaction queries via a XML-SCHEMA.

There are various types of sent messages:

1. **Transaction** type message (**simple**): offers information on a certain type of operation (e.g. Authorisation) of an order.
2. **Monitor** type message (**simple**): offers information on all types of operation (e.g. Authorisation and refund) associated with the same order number.
3. **Transaction** type message (**masivo**): offers a list of a certain type of operation generated on the merchant within a given timeframe.
4. **Monitor** type message (**masivo**): offers a list of all the operations of the merchant and terminal within a given timeframe.
5. Detail type message: Offers information on the details of a certain operation.

This is the XML-SCHEMA which the messages sent to the transaction query service must comply with:

```
<schema
targetNamespace="http://www.w3.org/namespace/"
xmlns:t="http://www.w3.org/namespace/"
xmlns="http://www.w3.org/2001/XMLSchema" elementForm-
Default="unqualified"
attributeFormDefault="unqualified">
  <element name="Messages">
      <complexType>
          <sequence>
              <element ref="t:Version"/>
              <element ref="t:Signature"/>
          </sequence>
      </complexType>
  </element>
  <element name="Version">
      <complexType>
          <sequence maxOccurs="unbounded">
              <element ref="t:Message"/>
          </sequence>
          <attribute name="Ds_Version" type="string"
          use="required"/>
      </complexType>
  </element>
  <element name="Message">
      <complexType>
          <choice>
              <element ref="t:Transaction"/>
              <element ref="t:Monitor"/>
              <element ref="t:Detail"/>
              <element ref="t:TransactionMasiva"/>
              <element ref="t:MonitorMasiva"/>
              <sequence minOccurs="0" maxOc-
              curs="unbounded">
              <element ref="t:Response"/>
              </sequence>
              <element ref="t:ErrorMsg"/>
          </choice>
      </complexType>
  </element>
  <element name="Transaction">
      <complexType>
          <sequence>
              <element ref="t:Ds_MerchantCode"/>
              <element ref="t:Ds_Terminal"/>
              <element ref="t:Ds_Order"/>
              <element ref="t:Ds_TransactionType"/>
              <element ref="t:Ds_Merchant_Data"
              minOccurs="0"/>
          </sequence>
      </complexType>
  </element>
  <element name="TransactionMasiva">
      <complexType>
          <sequence>
              <element ref="t:Ds_MerchantCode"/>
              <element ref="t:Ds_Terminal"/>
              <element ref="t:Ds_TransactionType"/>
              <element ref="t:Ds_Fecha_inicio"/>
              <element ref="t:Ds_Fecha_fin"/>
```

```
                </sequence>
            </complexType>
        </element>
        <element name="Monitor">
            <complexType>
                <sequence>
                    <element ref="t:Ds_MerchantCode"/>
                    <element ref="t:Ds_Terminal"/>
                    <element ref="t:Ds_Order"/>
                    <element ref="t:Ds_Merchant_Data"
                    minOccurs="0"/>
                </sequence>
            </complexType>
        </element>
        <element name="MonitorMasiva">
            <complexType>
                <sequence>
                    <element ref="t:Ds_MerchantCode"/>
                    <element ref="t:Ds_Terminal"/>
                    <element ref="t:Ds_Fecha_inicio"/>
                    <element ref="t:Ds_Fecha_fin"/>
                </sequence>
            </complexType>
        </element>
        <element name="Detail">
            <complexType>
                <sequence>
                    <element ref="t:Ds_MerchantCode"/>
                    <element ref="t:Ds_Terminal"/>
                    <element ref="t:Ds_Order"/>
                    <element ref="t:Ds_TransactionType"/>
                    <element ref="t:Ds_Merchant_Data"
                    minOccurs="0"/>
                </sequence>
            </complexType>
        </element>
        <element name="Response">
            <complexType>
                <sequence>
                    <element ref="t:Ds_MerchantCode"/>
                    <element ref="t:Ds_Terminal"/>
                    <element ref="t:Ds_Order"/>
                    <element ref="t:Ds_TransactionType"/>
                    <element ref="t:Ds_Date"/>
                    <element ref="t:Ds_Hour"/>
                    <element ref="t:Ds_Amount"/>
                    <element ref="t:Ds_Currency"/>
                    <choice minOccurs="0">
                        <sequence>
                            <element ref="t:Ds_CardNumber"/>
                            <element ref="t:Ds_ExpiryDate"/>
                        </sequence>
                        <element ref="t:Ds_TelephoneNumber"/>
                    </choice>
                    <element ref="t:Ds_SecurePayment"/>
                    <element ref="t:Ds_State"/>
                    <element ref="t:Ds_Response" minOc-
                    curs="0"/>
                    <element ref="t:Ds_Merchant_Data"
                    minOccurs="0"/>
                    <element ref="t:Ds_CardCountry" minOc-
                    curs="0"/>
                    <element ref="t:Ds_CardType" minOc-
                    curs="0"/>
                    <element ref="t:Ds_AuthorisationCode"
minOccurs="0"/>
                </sequence>
            </complexType>
        </element>
        <element name="Ds_MerchantCode">
            <simpleType>
                <restriction base="int">
                    <minInclusive value="1"/>
                    <maxInclusive value="999999999"/>
                </restriction>
            </simpleType>
        </element>
        <element name="Ds_Terminal">
            <simpleType>
                <restriction base="short">
                    <minInclusive value="1"/>
                    <maxInclusive value="999"/>
                </restriction>
            </simpleType>
        </element>
        <element name="Ds_Order">
            <simpleType>
                <restriction base="string">
                    <minLength value="1"/>
                    <maxLength value="12"/>
                </restriction>
            </simpleType>
        </element>
        <element name="Ds_TransactionType">
            <simpleType>
                <restriction base="string">
                    <length value="1"/>
                </restriction>
            </simpleType>
        </element>
        <element name="Ds_Merchant_Data" type="string"/>

        <element name="Ds_Fecha_fin" type="string"/>
        element name="Ds_Fecha_inicio" type="string"/>

        <element name="Ds_Date">
            <complexType mixed="true"/>
        </element>
        <element name="Ds_Hour">
            <complexType mixed="true"/>
        </element>
        <element name="Ds_Amount" type="long"/>
        <element name="Ds_Currency" type="short"/>
        <element name="Ds_CardNumber">
            <simpleType>
                <restriction base="string">
                    <minLength value="13"/>
                    <maxLength value="19"/>
                </restriction>
            </simpleType>
        element>
        <element name="Ds_ExpiryDate">
            <simpleType>
                <restriction base="string">
                    <length value="4"/>
                </restriction>
            </simpleType>
        </element>
        <element name="Ds_TelephoneNumber" type="int"/>
        <element name="Ds_SecurePayment">
            <simpleType>
                <restriction base="short">
                    <minInclusive value="0"/>
                    <maxInclusive value="1"/>
                </restriction>
            </simpleType>
```

```
  </element>
  <element name="Ds_State" type="string"/>
  <element name="Ds_Response" type="int"/>
  <element name="ErrorMsg">
        <complexType>
              <sequence>
                    <element ref="t:Ds_ErrorCode"/>

              </sequence>
        </complexType>
  </element>
  <element name="Ds_ErrorCode">
        <complexType mixed="true"/>
  </element>
  <element name="Signature" type="string"/>
  <element name="Ds_CardCountry">
        <simpleType>
              <restriction base="short">
                    <minInclusive value="1"/>
                    <maxInclusive value="999"/>
              </restriction>
        </simpleType>
  </element>
  <element name="Ds_CardType">
        <simpleType>
              <restriction base="string">
                    <length value="1"/>
              </restriction>
        </simpleType>
  </element>
  <element name="Ds_AuthorisationCode">
        <simpleType>
              <restriction base="string">
                    <minLength value="0"/>
                    <maxLength value="6"/>
              </restriction>
        </simpleType>
  </element>
</schema>
```

The possible fields which are sent in the requisition message maintain the format indicated upon creation of each message.

DATUM	NAME OF DATUM	Length / Type	COMMENTS
Merchant identification: FUC code	Ds_MerchantCode	9 / N	**Required.** Fuc code assigned to merchant.
Terminal number	Ds_Terminal	3 / No.	**Required.** Terminal number assigned by bank. 3 is considered the maximum length.
Order Number	Ds_Merchant_Order	12 / A-N	**Required.** The first 4 digits must be numerical; the remaining digits can only use the following ASCII characters: From 30 = 0 to 39 = 9 From 65 = to 90 = Z From 97 = to 122 = z
Type of transaction	Ds_TransactionType	1 / A-N	**Required field** for the merchant to indicate the type of transaction. Possible values are: 0 – Authorisation 1 – Pre-authorisation 4 – Reference payment 7 – Authentication A – Traditional payment
Merchant data	Ds_Merchant_Data	1024 / A-N	**Optional** field for the merchant to include in the data sent by the online response to the merchant if this option was selected.
Start Date	Ds_Fecha_inicio	26 / A-N	The date will have the format : yyyy-MM-dd-HH.mm.ss.xxxxxx An example for 1 December at 12:05 AM would be 2009-12-01-12.05.00.000000.
End Date	Ds_Fecha_fin	26 / A-N	The date will have the format : yyyy-MM-dd-HH.mm.ss.xxxxxx An example for 1 December at 12:05 AM would be 2009-12-01-12.05.00.000000.

We may find in the XML the fields enabling us to have the information requested in the query:

DATUM	NAME OF DATUM	Length / Type	COMMENTS
Merchant identification: FUC code	Ds_MerchantCode	9 / N	Required: Fuc code assigned to merchant.
Terminal number	Ds_Terminal	3 / N	Required: Terminal number assigned by bank. 3 is considered the maximum length.
Order Number	Ds_Order	12 / A-N	Required: The first 4 digits must be numerical; the remaining digits can only use the following ASCII characters: From 30 = 0 to 39 = 9 From 65 = to 90 = Z From 97 = to 122 = z
Type of transaction	Ds_TransactionType	1 / A-N	Required field for the merchant to indicate the type of transaction. Possible values are: 0 – Authorisation 1 – Pre-authorisation 4 – Reference payment 7 – Authentication A – Traditional payment
Merchant data	Ds_Merchant_Data	1024 / A-N	Optional field for the merchant to include in the data sent by the online response to the merchant if this option was selected.
Authorisation number	Ds_ AuthorisationCode	6 / A-N	Alphanumerical code of authorisation assigned to approval by authorising institution.
Date	Ds_Date	26 / A-N	The system date on which the operation occurred Example: 2009-09-14-11.59.59.999999.

Time	Ds_Hour	26 / A-N	The system time on which the operation occurred Example: 2009-09-14-11.59.59.999999.
Amount	Ds_Amount	No.	Amount of the operation.
Currency	Ds_Currency	3 / No.	Currency code The most common are : 978 : Euros 840 : Dollars 826 : Pounds 392 : Yen
Card No.	Ds_CardNumber	14-16 / No.	[Depends on merchant configuration]
Card type	Ds_CardType	1 / No.	C : Credit D : Debit - -> undetermined
Secure payment	Ds_SecurePayment	1 / No.	0 – If payment is NOT secure 1 – If payment is secure
Status	Ds_State	1 / A-N	Transaction status. Possible values: P: in progress F: finalised T: No response E: Operations with format error. S: Requested I: Special incident. W: Temporary status.
Error	Ds_ErrorCode	71 / A-N	Error code. (only in error message) See errors in errors table.
Response code	Ds_Response	4 / No.	See table below.

We may find in the XML the fields enabling us to have the information requested in the query:

CODE	MEANING
0000 to 0099	Transaction authorised for payments and pre-authorisations
0900	Transaction authorised for refunds and confirmations
101	Expired card
102	Card in transitory exceptional state or suspected fraud
104/9104	Operation not allowed for that card or terminal
116	Insufficient balance
118	Card not registered
129	Security code (CVV2/CVC2) incorrect
180	Non-service card
184	Error in holder authentication
190	Rejection without specifying Motive
191	Erroneous expiry date
202	Card in transitory exceptional state or suspected fraud with card withdrawal
912/9912	Issuer not available
Any other value	Transaction rejected

Note: Only in the case of pre-authentications (separate pre-authorisations), an 0 is return if it is authorised and the holder is authenticated and a 1 if it is authorised and the holder is not authenticated.

Based on this XML-SCHEMA, we can show various examples to check the input and output data of the XML.

The card field and expiry date will appear in accordance with the merchant configuration. In the example these will be shown, but are not required and do not appear unless specified in the configuration defined for the merchant.

1 - Simple Monitor

Message sent e.g:

```
<Messages>
    <Version Ds_Version="0.0">
    <Message>
        <Monitor>
            <Ds_MerchantCode>999008881</Ds_MerchantCode>
            <Ds_Terminal>1</Ds_Terminal>
            <Ds_Order>09102612333</Ds_Order>
            <Ds_Merchant_Data>El merchant data</Ds_Mer-
            chant_Data>
        </Monitor>
```

```
    </Mesage>
    </Version>
    <Signature>2a5fecc4f3d41274f5345503d580ac65dd-
    7be801</Signature>
</Messages>
```

Message sent e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <Response>
                <Ds_MerchntCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>09102613337</Ds_Order>
                <Ds_TransactionType>0</Ds_Transaction-
                Type>
                <Ds_Date>2009-10-26</D_Date>
                <Ds_Hour>12:33:37</Ds_Hour>
                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</DsCurrency>
                <Ds_CardNumber>4548810000000003</
                Ds_CardNumber>
                <Ds_CardType>C</Ds_Cardype>
                <Ds_ExpiryDate>1212</Ds_ExpiryDate>
                <Ds_SecurePayment>1</Ds_SecurePayment>
                <D_State>F</Ds_State>
                <Ds_Response>0</Ds_Response>
                <Ds_Merchant_Data>El merchant daa</
                Ds_Merchant_Data>
            </Response>
        </Message>
    </Version>
    <Signature>1eb3770ba531c7ecfcb557f623d-
    6b06a149c52f</Signature>
</Messages>
```

Nota: Los campos “Ds_CardNumber” y “Ds_ExpiryDate” dependen de la configuración del comercio.

2 - Simple transaction

Message sent e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <Transaction>
            <Ds_MerchantCode>999008881</Ds_Merchant-
            Code>
            <Ds_Terminal>1</Ds_Terminal>
            <Ds_Order>091026123337</Ds_Order>
            <Ds_TransactionType>0</Ds_TransactionType>
            <Ds_Merchant_Data>El merchant data</Ds_Mer-
            chant_Data>
            </Transaction>
        </Message>
    </Version>
    <Signature>f44ae1b30659c3441b52c1cd0e-
    14f55ae4c76082</Signature>
</Messages>
```

Response message e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>091026123337</Ds_Order>
                <Ds_TransactionType>0</Ds_Transaction-
                Type>
                <Ds_Date>2009-10-26</Ds_Date>
                <Ds_Hour>12:33:37</Ds_Hour>
                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>
<Ds_CardNumber>4548810000000003</Ds_CardNumber>
                <Ds_CardType>C</Ds_CardType>
                <Ds_ExpiryDate>1212</Ds_ExpiryDate>
                <Ds_SecurePayment>1</Ds_SecurePay-
                ment>
                <Ds_State>F</Ds_State>
                <Ds_Response>0</Ds_Response>
                <Ds_Merchant_Data>El merchant data</
                Ds_Merchant_Data>
            </Response>
        </Message>
    </Version>
    <Signature>1eb3770ba531c7ecf0cb557f623d-
    6b06a149c52f</Signature>
</Messages>
```

NOTA: Los campos “Ds_CardNumber” y “Ds_ExpiryDate” dependen de la configuración del comercio.

3 - Mass Monitor

Message sent e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <MonitorMasiva>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Fecha_inicio>
                2009-09-10-00.00.00.000000
                </Ds_Fecha_inicio>
                <Ds_Fecha_fin>
                2009-09-14-11.59.59.999999
                </Ds_Fecha_fin>
            </MonitorMasiva>
        </Message>
    </Version>
    <Signature>7b1df6940c00271a2f47fb-
    2de2353487f6430066
    </Signature>
</Messages>
```

Response message e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>090910132731</Ds_Order>
                <Ds_TransactionType>0</Ds_Transaction-
```

```
        Type>
        <Ds_Date>2009-09-10</Ds_Date>
        <Ds_Hour>13:27:32</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>S</Ds_State>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>090910135448</Ds_Order>
        <Ds_TransactionType>1</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-10</Ds_Date>
        <Ds_Hour>13:55:11</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548810000000003</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>F</Ds_State>
        <Ds_Response>0</Ds_Response>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>090911113431</Ds_Order>
        <Ds_TransactionType>0</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-11</Ds_Date>
        <Ds_Hour>11:34:32</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>S</Ds_State>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>090911113550</Ds_Order>
        <Ds_TransactionType>0</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-11</Ds_Date>
        <Ds_Hour>11:35:51</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>S</Ds_State>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>090911113851</Ds_Order>
        <Ds_TransactionType>0</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-11</Ds_Date>
        <Ds_Hour>11:38:52</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548030000000008</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>A</Ds_State>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>090911114406</Ds_Order>
        <Ds_TransactionType>0</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-11</Ds_Date>
        <Ds_Hour>11:44:43</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548030000000008</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>A</Ds_State>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>090911114836</Ds_Order>
        <Ds_TransactionType>0</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-11</Ds_Date>
        <Ds_Hour>11:48:37</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548030000000008</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>A</Ds_State>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>090914090801</Ds_Order>
        <Ds_TransactionType>0</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-14</Ds_Date>
        <Ds_Hour>09:08:02</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4940198000000007</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>1212</Ds_ExpiryDate>
```

```
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>A</Ds_State>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>1234567890</Ds_Order>
        <Ds_TransactionType>1</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-10</Ds_Date>
        <Ds_Hour>13:27:40</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548810000000003</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>F</Ds_State>
        <Ds_Response>0</Ds_Response>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>1234567890</Ds_Order>
        <Ds_TransactionType>2</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-10</Ds_Date>
        <Ds_Hour>13:28:28</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548810000000003</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>F</Ds_State>
        <Ds_Response>900</Ds_Response>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>1234567890</Ds_Order>
        <Ds_TransactionType>3</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-10</Ds_Date>
        <Ds_Hour>13:29:53</Ds_Hour>
        <Ds_Amount>145</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548810000000003</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>F</Ds_State>
        <Ds_Response>900</Ds_Response>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>1234567890</Ds_Order>
        <Ds_TransactionType>3</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-10</Ds_Date>
        <Ds_Hour>13:30:43</Ds_Hour>
        <Ds_Amount>1</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548810000000003</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>F</Ds_State>
        <Ds_Response>900</Ds_Response>
    </Response>
    <Response>
        <Ds_MerchantCode>999008881</Ds_Mer-
        chantCode>
        <Ds_Terminal>1</Ds_Terminal>
        <Ds_Order>1234567890</Ds_Order>
        <Ds_TransactionType>3</Ds_Transaction-
        Type>
        <Ds_Date>2009-09-10</Ds_Date>
        <Ds_Hour>13:38:00</Ds_Hour>
        <Ds_Amount>2</Ds_Amount>
        <Ds_Currency>978</Ds_Currency>

        <Ds_CardNumber>4548810000000003</
        Ds_CardNumber>
        <Ds_CardType>null</Ds_CardType>
        <Ds_ExpiryDate>0909</Ds_ExpiryDate>
        <Ds_SecurePayment>0</Ds_SecurePay-
        ment>
        <Ds_State>F</Ds_State>
        <Ds_Response>900</Ds_Response>
    </Response>
  </Message>
 </Version>
 <Signature>44c8a04b33c4feeb8bc5bb-
 b879626307586244c4</Signature>
</Messages>
```

Nota: Los campos “Ds_CardNumber” y “Ds_ExpiryDate” dependen de la configuración del comercio.

4 - Mass transaction

Message sent e.g.:

```
<Messages>
    <Version Ds_Version=“0.0”>
        <Message>
            <TransactionMasiva>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_TransactionType>0</Ds_Transaction-
                Type>
                <Ds_Fecha_ini-
                cio>2009-09-10-00.00.00.000000</Ds_Fe-
                cha_inicio>
                <Ds_Fecha_fin>2009-09-14-
```

```
                11.59.59.999999</Ds_Fecha_fin>
            </TransactionMasiva>
        </Message>
    </Version>
    <Signature>f40a4de448f4539a423582d02be1303bd86c-
    f4a0</Signature>
</Messages>
```

Response message e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>090911113431</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Date>2009-09-11</Ds_Date>
                <Ds_Hour>11:34:32</Ds_Hour>
                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>
                <Ds_SecurePayment>0</Ds_SecurePay-
                ment>
                <Ds_State>S</Ds_State>
            </Response>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>090911113550</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Date>2009-09-11</Ds_Date>
                <Ds_Hour>11:35:51</Ds_Hour>
                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>
                <Ds_SecurePayment>0</Ds_SecurePay-
                ment>
                <Ds_State>S</Ds_State>

            </Response>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>090911113851</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Date>2009-09-11</Ds_Date>
                <Ds_Hour>11:38:52</Ds_Hour>
                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>

                <Ds_CardNumber>4548030000000008</
                Ds_CardNumber>
                <Ds_CardType>null</Ds_CardType>
                <Ds_ExpiryDate>0909</Ds_ExpiryDate>
                <Ds_SecurePayment>0</Ds_SecurePay-
                ment>
                <Ds_State>A</Ds_State>
            </Response>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>090911114406</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Date>2009-09-11</Ds_Date>
                <Ds_Hour>11:44:43</Ds_Hour>
                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>

                <Ds_CardNumber>4548030000000008</
                Ds_CardNumber>
                <Ds_CardType>null</Ds_CardType>
                <Ds_ExpiryDate>0909</Ds_ExpiryDate>
                <Ds_SecurePayment>0</Ds_SecurePay-
                ment>
                <Ds_State>A</Ds_State>
            </Response>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>090911114836</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Date>2009-09-11</Ds_Date>
                <Ds_Hour>11:48:37</Ds_Hour>

                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>

                <Ds_CardNumber>4548030000000008</
                Ds_CardNumber>
                <Ds_CardType>null</Ds_CardType>
                <Ds_ExpiryDate>0909</Ds_ExpiryDate>
                <Ds_SecurePayment>0</Ds_SecurePay-
                ment>
                <Ds_State>A</Ds_State>
            </Response>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>1</Ds_Terminal>
                <Ds_Order>090914090801</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Date>2009-09-14</Ds_Date>
                <Ds_Hour>09:08:02</Ds_Hour>
                <Ds_Amount>145</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>

                <Ds_CardNumber>4940198000000007</
                Ds_CardNumber>
                <Ds_CardType>null</Ds_CardType>
                <Ds_ExpiryDate>1212</Ds_ExpiryDate>
                <Ds_SecurePayment>0</Ds_SecurePay-
                ment>
                <Ds_State>A</Ds_State>
            </Response>
        </Message>
    </Version>
    <Signature>a5a38ed7336808b7695363ebec-
    2879e2e419e7b4</Signature>
</Messages>
```

Nota: Los campos “Ds_CardNumber” y “Ds_ExpiryDate” dependen de la configuración del comercio.

5 - Detail transaction

Message sent e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
```

```
            <Detail>
                <Ds_MerchantCode>999008881
                </Ds_MerchantCode>
                <Ds_Terminal>16</Ds_Terminal>
                <Ds_Order>100511114713</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Merchant_Data>El merchant data</
                Ds_Merchant_Data>
            </Detail>
        </Message>
    </Version>
    <Signature>5b808cc7cbfea755600ff226b22107307d-
    d11d29</Signature>
</Messages>
```

Response message e.g.:

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <Response>
                <Ds_MerchantCode>999008881</Ds_Mer-
                chantCode>
                <Ds_Terminal>16</Ds_Terminal>
                <Ds_Order>100511114713</Ds_Order>
                <Ds_TransactionType>0</Ds_TransactionType>
                <Ds_Date>2010-05-11</Ds_Date>
                <Ds_Hour>11:47:15</Ds_Hour>
                <Ds_Amount>12244</Ds_Amount>
                <Ds_Currency>978</Ds_Currency>
                <Ds_SecurePayment>1</Ds_SecurePayment>
                <Ds_State>F</Ds_State>
                <Ds_Response>0</Ds_Response>
                <Ds_Merchant_Data>El merchant data</
                Ds_Merchant_Data>
                <Ds_AuthorisationCode>050197</Ds_Au-
                thorisationCode>
            </Response>
        </Message>
    </Version>
    <Signature>1bb5122958b0c4234df5f-
    15d3e8a1c1961ec8626</Signature>
</Messages>
```

6 - Example of response with an Error:

In this case, a signature error.

```
<Messages>
    <Version Ds_Version="0.0">
        <Message>
            <ErrorMsg>
                <Ds_ErrorCode>SIS0034</Ds_ErrorCode>
            </ErrorMsg>
        </Message>
    </Version>
</Messages>
```

8.3 WSDL of the service

The URL's of the Virtual POS WEB services are as follows:

Test environment:
https://sis-t.redsys.es:25443/apl02/service

Real environment:
https://sis.redsys.es/apl02/services/SerClsWSConsulta

These URL se will be used as the service destination point.

The URLs with the available WSDL are at these addresses:

Test environment:
https://sis-t.redsys.es:25443/apl02/services/SerClsWSConsulta/wsdl/SerClsWSConsulta.wsdl

Real environment:
https://sis.redsys.es/apl02/services/SerClsWSConsulta/wsdl/SerClsWSConsulta.wsdl

The WSDL which describes the transaction query service of the SIS is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
    name="SerClsConsultasSIS"
    targetNamespace="http://tempuri.org/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xs-
    d=http://www.w3.org/2001/XMLSchema
    xmlns:tns=http://tempuri.org/
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

<wsdl:message name="procesaMensajeRecibidoResponse">
    <wsdl:part name="return" type="xsd:string"/>
</wsdl:message>

<wsdl:message name="procesaMensajeRecibidoRequest">
    <wsdl:part name="Mensaje" type="xsd:string"/>
</wsdl:message>

<wsdl:portType name="SerClsConsultasSISPortType">
    <wsdl:operation name="procesaMensajeRecibido">
        <wsdl:input message="tns:procesaMensajeRecibidoRe-
        quest"/>
        <wsdl:output
        message="tns:procesaMensajeRecibidoResponse"/>
    </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="SerClsConsultasSISBinding"
    type="tns:SerClsConsultasSISPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.
org/soap/http"/>
        <wsdl:operation name="procesaMensajeRecibido">
        <soap:operation soapAction="urn:mensajeriaCIBER-
        PAC#procesaMensajeRecibido"/>
<wsdl:input>
<soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:mensajeriaCIBERPAC"/>
```

```
    </wsdl:input>
    <wsdl:output>
<soap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/
        encoding/"
        namespace="urn:mensajeriaCIBERPAC"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>

<wsdl:service name="SerClsConsultasSISService">
<wsdl:port name="SerClsConsultasSISPort"
    binding="tns:SerClsConsultasSISBinding">
    <soap:address location="https://sis.redsys.es/apISOAP/
    rpcrouter"/>
</wsdl:port>
</wsdl:service>

</wsdl:definitions>
```

8.4 Example of SOAP client

Below is an example of a SOAP client made in JAVA. The SOAP client must effect the following steps:

- Indicate the SOAP service URL to be accessed

  URL url = new URL ("https://sis.redsys.es/apl02/services/SerClsWSConsulta ");

- Create a SOAPMappingRegistry type object:

  SOAPMappingRegistry smr = new SOAP-MappingRegistry();

  Create a Call type object with the following data:

  SOAPMappingRegistry (previously created)

  TargetObjectURI. Urn of the messaging service.

  MethodName. Method to be accessed.

  EncodingStyleURI. Constant.

- Vector with query parameters

For example:

```
Call call = new Call();
call.setSOAPMappingRegistry(smr);
call.setTargetObjectURI("urn:mensajeriaCIBERPAC");
call.setMethodName("procesaMensajeRecibido");
call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
Vector params = new Vector();
params.addElement(new Parameter("Mensaje", String.
class, xml_doc, null));
call.setParams(params);
-
Realizar invoke con la URL del servicio SOAP, que retornará
un objeto "Response"
Response resp=null;
resp = call.invoke(url, "");
Parameter ret = resp.getReturnValue();
Object value = ret.getValue();
```

EXAMPLE OF SOAP JAVA CLIENT (SERVLET)

```
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

import java.io.*;
Import java.net. *;
import org.apache.soap.*;
import org.apache.soap.messaging.*;
import org.apache.soap.transport.*;
import org.apache.soap.util.xml.*;
import org.apache.soap.encoding.*;
import org.apache.soap.encoding.soapenc.*;
import org.apache.soap.rpc.*;

public class SerSvlCIBERPAC extends HttpServlet
{
    public void doPost(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException
    {
        String respuesta = "";
    try
    {

        String xml_doc = req.getParameter("elXMLEnvio");
        respuesta = ejecutaCallRPCStyle(xml_doc);
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}

    public String ejecutaCallRPCStyle(String xml_doc) throws
ServletException, IOException
    {
        String respuesta = "";

        String encodingStyleURI = Constants.NS_URI_SOAP_ENC;
        URL url = new URL("https://sis.redsys.es/apISOAP/
        rpcrouter");
        SOAPMappingRegistry smr = new SOAPMappingRegistry();
        Call call = new Call();
        call.setSOAPMappingRegistry(smr);
        call.setTargetObjectURI("urn:mensajeriaCIBERPAC");
        call.setMethodName("procesaMensajeRecibido");
        call.setEncodingStyleURI(encodingStyleURI);
        Vector params = new Vector();
        params.addElement(new Parameter("Mensaje", String.class,
        xml_doc, null));
        call.setParams(params);
        Response resp=null;
        try
        {
            resp = call.invoke(url, "");
```

```
}
catch (SOAPException e)
{
    e.printStackTrace();
}
if (!resp.generatedFault())
{
    Parameter ret = resp.getReturnValue();
    Object value = ret.getValue();
    respuesta = (String) value;
}
else
{
        Fault fault = resp.getFault();
        respuesta = fault.getFaultString();
    }
    return (respuesta);
  }
}
```

8.5 SOAP error codes

The SOAP transaction query service has its own error codes. They are as follows.

ERROR	DESCRIPTION
XML0000	Sundry errors in processing XML-String received.
XML0001	Error generating the DOM from the XML-String received and the DTD defined.
XML0002	Error "Message" element does not exist in XML-String received.
XML0003	"Message" type error in the XML-String received has an unknown or invalid value in the requisition.
XML0004	Error "Ds MerchantCode" element does not exist in XML-String received.
XML0005	Error the "Ds_MerchantCode" element in the XML-String received is empty.
XML0006	Error the "Ds_MerchantCode" element has an incorrect length in the XML-String received.
XML0007	Error the "Ds_MerchantCode" element has an incorrect length in the XML-String received.
XML0008	Error "Ds_Terminal" element does not exist in XML-String received.

XML0009	Error the "Ds_Terminal" element in the XML-String received is empty.
XML0010	Error the "Ds_Terminal" element has an incorrect length in the XML-String received
XML0011	Error the "Ds_Terminal" element has an incorrect numerical format in the XML-String received.
XML0012	Error "Ds_Order" element does not exist in XML-String received.
XML0013	Error the "Ds_Order" element in the XML-String received is empty.
XML0014	Error the "Ds_Order" element has an incorrect length in the XML-String received.
XML0015	Error the "Ds_Order" element does not have its first 4 numerical positions in the XML-String received.
XML0016	Error "Ds_TransactionType" element does not exist in XML-String received.
XML0017	Error the "Ds_TransactionType" element in the XML-String received is empty.
XML0018	Error the "Ds_TransactionType" element has an incorrect length in the XML-String received.
XML0019	Error the "Ds_TransactionType" element has an incorrect numerical format in the XML-String received.
XML0020	Error the "Ds_TransactionType" element has an unknown or invalid value in a Transaction message.
XML0021	Error "Signature " element does not exist in XML-String received.
XML0022	Error the "Signature" element in the XML-String received is empty.

XML0023	Error the signature is not correct.
XML0024	There are no operations in TZE for the data requested.
XML0025	Error the response XML is incorrectly formed.
XML0026	Error "Ds_fecha_inicio" element does not exist in XML-String received.
XML0027	Error "Ds_fecha_fin " element does not exist in XML-String received.

9.
Periodic information files

Banco Sabadell has created a file system with information on credit and debit card operations carried out at Merchants. They are the following types of files:

- Operations file
- Charge-backs file
- Confirmed fraud file
- Documentation requisition file

The files are generated every day from Monday to Friday provided they are not national holidays.

Operations file

This file provides information on the operations processed by the merchant and which have been posted in its financial account.

Operations processed during the day are not described; only operations settled.

The sum of the operations listed matches the amount paid daily into the merchant account.

Charge-backs file

Charge-backs of operations are the cardholder's method for requesting, via their financial entity (issuing bank) the refund of the purchase. These charge-backs must conform to the regulations of the relevant card (Visa or MasterCard).

Banco Sabadell verifies the conformity of chargebacks received to said regulations, directly representing the issuing bank in those which it does not consider legitimate and charging the merchant account those which are accepted. In some cases Banco Sabadell may have to request documentation from the establishment either to evaluate the possibility of assuming representation or to document this representation for greater possibility of success. In these cases the merchant will receive a documentation requisition to be answered as described in section 5.2 of this manual.

Banco Sabadell supplies information on the chargebacks received, whether or not charged to the account, so the merchant knows which customers have requested them. Thus, the merchant's security department can contact them, cancel their subscriptions, add them to blacklists or any other measure it deems necessary. This file is therefore not for accounting purposes but is merely for informative purposes.

Confirmed fraud file

Confirmed fraud is the tool the issuing bank uses to notify a fraudulent operation (operation not performed or authorised by the cardholder).

This modification is totally independent of the existence of a previous or subsequent chargeback, or of whether it has been charged to the merchant or been represented. This is a method for notifying that the transaction was not performed by the cardholder, so that Visa/MasterCard can detect those merchants that process a high rate of fraudulent operations.

A high rate of confirmed fraud is an indication that the merchant is carrying on a fraudulent activity or is suffering an attack by illicit purchasers and has not put in place the necessary systems to reduce it.

If this level is far in excess of that allowed or occurs over several consecutive months, the card companies demand cancellation of the contract with the establishment and occasionally impose substantial economic penalties on the latter.

In the same way as the chargeback file, Banco Sabadell generates a daily file with the Confirmed Fraud operations which have been reported by the various card companies. This file, like the previous one, is purely for

information purposes and has no book data. Its purpose is to inform the establishment of those cards which are being used for fraud so that their security department can contact the customers affected, cancel their accounts, add them to blacklists or take any other appropriate measure.

Documentation requisition file.

The requisition of documentation is an option the cardholder has and the issuing bank to check the validity of an operation either because they do not remember the purchase, because they do not recognise it or any other reason.

Anyone making a purchase (presential or otherwise) using a financial card has the right to request the merchant provide the documentation proving payment. This request is usually due to the card owner not remembering the operation or wishes to have more details or argues that they did not perform the operation and attempt to reverse it. In certain cases it is because they do not associate the merchant name with the webpage on which the operation was carried out.

This documentation is requested to carry out the defence or representation of an operation and must be compulsorily provided to avoid a chargeback

The merchant is under the obligation to submit it within 7 Merchant Days in accordance with the instructions described in section 5.2 of this manual.

9.1 Channels for receiving files

Files can be obtained via various channels:

- BS Online
- FTP

BS Online

The merchant connects to the Banco Sabadell Group home banking service and downloads the file which contains the operations settled since the last download.

If it is not downloaded on one day, the operations accumulate for the next.

To obtain this information the merchant must have the Distance Banking Service..

FTP

Once the daily process of payment to merchants by the Banco Sabadell Group has finalised, the files are available on an accessible resource.

The client connects and transfers them to their system using FTPS (File Transfer Pro¬tocol Secure) or EDITRAN or SWIFTNet sessions are established to transfer the fles generated.

To access this information, the customer must request it from his branch and provide the following details:

- Technology contact person
- Telephone no.
- Email
- IP of client system
- Communication Channel (FTPS, EDITRAN, SWIFTNet,...).

EDITRAN

Once the daily process of payment to merchants by the Banco Sabadell Group has finalised, the “specific” files for certain clients are available in EDITRAN/BS.

If required, the merchant can receive the file automatically on their system or collect it when desired.

To access this information, the client must request it from his branch and provide the following details for the Banco Sabadell Group

technicians to contact the technology supervisor of the merchant:

- Technology contact person
- Telephone no.
- Email
- IP of client system

9.2 Description of content of the files

Below we describe the contents of each of the files available:

1. Operations file

File header register

DATA	FORMAT	DESCRIPTION
Register type	XX	10
Date processed	DD-MM-YYYY	Settlement processing date
Start Date	DD-MM-YYYY	Date on which operations entered
End Date	DD-MM-YYYY	Date until which operations entered
Fill-in	X(188)	

Remittance header register

DATA	FORMAT	DESCRIPTION
Register type	XX	00
Contract	9(15)	Merchant contract number
Merchant	9(10)	FUC number of the merchant
Account	9(20)	CAC of merchant account
Office	X(4)	Merchant management branch
Date processed	DD-MM-YYYY	Settlement processing date
Start Date	DD-MM-YYYY	Date on which operations entered

DATA	FORMAT	DESCRIPTION
End Date	DD-MM-YYYY	Date until which operations entered
Fill-in	X(139)	

Log of operations.

DATA	FORMAT	DESCRIPTION
Register type	XX	01
Value	DD-MM-YYYY	Payment valuation date
Remittance	9(10)	Remittance number
Invoice	X(12)	Invoice number of the operation → Invoice number of the original operation if a Chargeback
Remittance branch	9(4)	Remittance branch
Card	999999++++++9999 999999+++++++9999 999999++++++++9999 999999+++++++++9999	Card number
Card brand	X	V - Visa M - Master (includes Maestro) O - Other
Card type	X	E - Company P - Individuals B - B2B
Type of payment	X	C - Credit D - Debit
Card issuing entity	X	P - Banco Sabadell N – National Servired system R – Other national systems E – International euro zone I - International others: O - Other
Operation date	DD-MM-AAAA	Date of operation
Operation time	HHMMSS	Time of the operation.

DATA	FORMAT	DESCRIPTION
Authorisation	9(6)	Authorisation number
Type of operation	XX	5 - Sale (+) / Cancellation of sale (-) 6 – Total or partial refund (-) / Cancellation of total or partial refund (+) 15 – Chargeback of refund (+), ordered by cardholder 16 - Chargeback or return of sale (-), ordered by cardholder
Operation capture	XXX	ON - Online OFF - Offline
Amount of Operation	9 Integers 2 decimals	Nominal amount of the operation
Operation sign	X	+ Credit - Charge
Discount rate	3 Integers 2 decimals	Discount percentage
Discount amount	7 Integers 2 decimals	Nominal amount - Discount.
Discount sign	X	+ Credit - Charge
Net amount	11 Integers 2 decimals	Nominal amount - Discount.
Net sign	X	+ Credit - Charge
POS	9(13)	PoS number
ARN	X(23)	Acquirer / Purchase Reference Number
Fill-in	X(9)	
Fill-in	X(5)	
Fill-in	X(01)	
Merchant currency	X(3)	Merchant settlement currency
Number of operation	9(12)	Number of operation
Info Code	X(4)	Chargeback info code

DATA	FORMAT	DESCRIPTION
Fill-in	XX	00
Original amount	11 Integers 2 decimals	Amount in original currency (if other than euro)
Original sign	X	+ Credit - Charge
Original currency	X(3)	Currency of settlement for cardholder
Fill-in	X	00

Merchant queue register

DATA	FORMAT	DESCRIPTION
Register type	XX	99
Fill-in	X(25)	
Operations	9(9)	Number of merchant operations
Amount	11 Integers 2 decimals	Total amount of merchant operations
Sign	X	+ Credit - Charge
Fill-in	X(170)	

File queue register

DATA	FORMAT	DESCRIPTION
Register type	XX	90
Merchants	9(9)	Number of merchants in file
Fill-in	X(25)	
Operations	9(9)	Number of operations in file

DATA	FORMAT	DESCRIPTION
Amount	11 Integers 2 decimals	Total amount of operations: of file
Sign	X	+ Credit - Charge
Fill-in	X(161)	

2. Charge-backs file

The file contains a single type of register with the data separated by a semicolon (;) whose format is as follows:

DATA	DESCRIPTION
Register type	CB
Merchant	FUC number of the merchant
Reception date	Charge-back reception date
Card	999999++++++9999 999999+++++++9999 999999++++++++9999 999999+++++++++9999
Chargeback amount	Amount charged back
Chargeback currency	Chargeback currency
Operation date	Original Operation date
Operation time	Time of Original Operation
Remittance date	Original Operation remittance date
Remittance	Original Operation remittance number
Invoice	Original Operation bill number
Amount of Operation	Amount of Original Operation

DATA	DESCRIPTION
Operation currency	Currency of Original Operation
Number of operation	Number of Original Operation
Incident type	15 - Chargeback or sale return (-), ordered by cardholder 16 - Chargeback or sale return (-), ordered by cardholder
Info Code	Info code of chargeback (see table)
Order number	Incident number
Text	Text accompanying chargeback received
ARN	Acquirer / Purchase Reference Number X(23)

Table of REASON CODE for chargeback

CODE REASON	VISA	MASTERCARD	DESCRIPTION
4501	78		Invalid transaction
4503	73	4835	Card expired
4506	80		Error when processing.
4507			Amount of transaction incorrect
4510		4850	Credit processed as debit
4512	82	4834	Transaction processed more than once
4513	85	4860	Credit not received
4514	93	4849	Fraudulent transaction (AMD)
4515		4515	Transaction not completed
4516		4801	Request for receipt with confirming dispatch
4517		4802	

CODE REASON	VISA	MASTERCARD	DESCRIPTION
4521	72	4808	Amount > Limit not authorised
4522	71	4522	Operation rejected.
4523		4812	Inexistent card
4524		4831	Error in sum
4525	86		Paid by other means
4526		4837	No signature.
4527	81		No card printout
4530			Errors in currency conversion
4531		4854	Dispute of unclassified holder. In other CDG
4532			Faulty goods
4534			Printout of multiple receipts
4535		4807	Card included in exceptions file
4536	74	4842	Presented after deadline
4537			Dispute in presenting reserve transfer
4538			Deposit for reservation of accommodation
4540			Card not present
4544	41	4841	Recurring operation cancelled
4545	76		Currency conversion not allowed
4546	83		Fraud in non-presential environment
4547	70		Card included in bulltin
4549	77		Incorrect card number
4550		4857	Holder does not recognise operation
4551		4846	Transaction currency not entered / erroneous
4553			Goods not initially contracted
4554	90	4855	Goods not received

CODE REASON	VISA	MASTERCARD	DESCRIPTION
4555	30	4859	Services not provided (U.S)
4556	53	4853	Goods other than specified
4557	3		Merchant without CAE in fuel operation
4703			Dispute on Ebir adjustment
4728			Cancelled pre-authorization
4750			Car hire Non-justified charge
4751			Authorisation expired
4752			Credit/debit Error in presentation
4753		4809	Operation not reconciled
4754			Legal regulation/legal dispute
4755			Invalid holder authorisation code
4757			Goods not sent to address given
4758			Expiry date not yet provided
4759			TC and relevant calc. cannot be
4762	62	4862	Magnetic stripe of card forged
4763			Recourse complete
4791	91		Elec. Card data incorrectly entered
4792			Fees refund for unsuccessful balance inq
4793			Prohibited Merchant
4798			Card verification fails
4803		4803	First chargeback not in historic file
4804		4804	Mds Regulation:

CODE REASON	VISA	MASTERCARD	DESCRIPTION
4857	57	4840	Multiple non-authorised transactions
4863	75	4863	Transaction not acknowledged by holder
4870		4870	EMV LAIBILITY SHIFT- Forgery
4871		4871	EMV LAIBILITY SHIFT – Card lost/ stolen
4880		4880	Chip POS Operation Late presentation
4896	96		Exceeds limit (terminal amount. Limited)
4899		4899	Operation not allowed
4901			Documentation required not received in representation
4902			Doc. Received in representation illegible
4903			Doc. Received in repr. Invalid or incomplete
4905			Ref. Data Adq invalid doc not received
4908		4847	Exceeds limit (unauthorised operation)
4918			Ref. Data Adq invalid doc received
4924			Card included in bulletin
4999		4999	Dispute in chargeback in Europe

3. Confirmed fraud file

The file contains a single type of register with the data separated by a semicolon (;) whose format is as follows:

DATA	DESCRIPTION
Register type	FC
Merchant	FUC number of the merchant
Report date	Report date of confirmed fraud
Card	999999++++++9999 999999+++++++9999 999999++++++++9999 999999+++++++++9999
Fraud Amount	Amount of fraud
Fraud currency	Fraud currency
Operation date	Original Operation date
Operation time	Time of Original Operation
Remittance date	Original Operation remittance date
Remittance	Original Operation remittance number
Invoice	Original Operation bill number
Amount of Operation	Amount of Original Operation
Operation currency	Currency of Original Operation
Number of operation	Number of Original Operation
ARN	Acquirer / Purchase Reference Number X(23)

4. Documentation request file

This file contains a single type of register whose format is as follows:

DATA	FORMAT	DESCRIPTION
Date processed	DD-MM-AAAA	Day Processed
FUC	9(10)	FUC number of the merchant
Merchant	X(10)	Reduced merchant name
Telephone	9(9)	Mrchant telephone number
Fill-in	XX	
Settlement date	DD-MM-AA	Settlement date
Fill-in	X	
Remittance	9(10)	Remittance number
Fill-in	X	
Invoice	9(12)	Bill number
Fill-in	X(4)	
Operation date	DD-MM-AA	Date of operation
Fill-in	X	
Card	999999++++++9999 999999+++++++9999 999999++++++++9999 999999+++++++++9999	Card number
Amount of Operation	11 Integers 2 decimals	Amount of the operation
Currency	X	Currency (E = euro)
.	X(30)	Additional information
Fill-in	X(10)	
ARN	X(23)	Acquirer / Purchase Reference Number

10.
Supervision programs and penalties

The Visa and MasterCard card companies have supervision and control programs for chargebacks received by merchants. These programs establish the relevant penalties for those merchants which exceed the maximum percentage of chargebacks allowed.

Penalties will be applied to merchants which exceed 50 chargeback operations during two consecutive months, the percentage of refunds of charged received exceeds 1% of transactions of the previous month.

In addition, programs have been put in place to identify those merchants which, despite not exceeding the above parameters which give rise to the imposition of penalties, receive ratios close to the operations charged back.

The alert parameters are:

– Merchants which exceed 50 chargeback operations and the percentage of refunds of charges exceeds 0.5% of transactions of the previous month.

The alert parameters do not necessarily signify that the merchant will be penalised, but compel the merchant to provide explanations and send an action plan to demonstrate that it is taking all the necessary measures to avoid high volumes of charged back operations being repeated in future.

If the Action Plan sent by the merchant is considered insufficient by the card companies, the latter may request a penalty be imposed consisting of the temporary suspension of up to one month in processing cards.

If a merchant appears on several occasions in an alert file, more severe action may be taken, including exclusion of the merchant from the card payment gateway. For this reason, despite not direct penalty existing, it is necessary to take into account the maximum alert parameters as the maximum acceptable values for the merchant.

11.
PCI DSS - Card data security program

11.1 What is PCI DSS?

Payment Card Industry Data Security Standard (PCI DSS) es:

- **A security standard** whose aim is to protect the card data in any location.
- It is **mandatory** for any company that **stores, processes or transfers** card data.
- Promoted by the leading international brands of cards which, in order to create it, set up the **PCI Security Standard Council** in 2006 (Visa, MasterCard, American Express, JCB and Discover), with all the information related to this standard contained and updated on its website: www.pcisecuritystandards.org.

Before starting to integrate your Virtual PoS, we recommend you read Best Practices for Securing E-commerce.

11.2 What are the card data in accordance with PCI DSS?

In the eCommerce environment, the card data to be protected is:

- Name of the cardholder
- Card number or PAN (Primary Account Number)
- **Expiry date**
- The **CVV2, CVC2, CAV, CID2**, etc.

The PAN and other data may be stored provided the PAN is rendered illegible, in compliance with any of the systems established by the PCI DSS.

The CVV2, CVC2, etc., can never be stored following authorisation, even when encrypted.

11.3 Objective of the PCI DSS

a. Guarantee protection of cardholder information.

b. Minimize the risk of possible non-authorised intrusions or compromising of account and card information.

c. Improve the level of security of card payments, furthering the existence of a secure payment environment for the information.

d. Combat identity theft and other frauds present on the Internet.

e. Increase cardholders' trust in card transactions.

11.4 Advantages for merchants

a. Promote integrity of the merchant and increase consumer trust in the business.

b. Increase sales as a result of increased trust by consumers.

c. Protect the merchant from possible loss of income, undesired investigations and legal costs.

d. Reduce the reputational risk of the merchant as a result of a leak of customer information.

e. Reduce disputes with cardholders and the costs associated with fraudulent transactions resulting from information leaks.

f. Prevent mass theft of customer information.

g. Facilitate the adoption of security standards which are globally valid.

11.5 Who must comply with PCI DSS?

It is mandatory for any company that stores, processes or transfers card data.

11.6 Which companies must validate compliance and how is this done?

On the basis of potential fraud risk criteria, associated with the possession of card

data, the international card brands Visa and MasterCard classify merchants and suppliers by levels and establish different formulas to validate compliance.

LEVEL	CRITERIA	METHOD OF COMPLIANCE VALIDATION	SUPERVISORY ENTITY
1	Any merchant processing **over six million in annual transactions** regardless of the channel. Any merchant or supplier that has suffered a confirmed security incident. **Service providers** (and TPP) that process **over 300,000 card transactions a year**.	Annual Audit by a QSA (Quality Security Assessor) Quarterly scanning of network vulnerability with an ASV (Approved Scanning Providers)	Independent security consultant or own company if signed by a representative of the security company. Specialist from the security company.
2	Any merchant processing **over 1 million and fewer than 6 million annual transactions** regardless of the channel. **Service providers** (except TPP) that process **under 300,000 card transactions a year**.	Annual self-evaluation questionnaire (SAQ)	Merchant itself.
3	Any merchant processing **over 20.000 and fewer than 1 million annual transactions** regardless of the channel.		
4	Rest: **Any merchant not included in the above levels.** Level 4 merchants obliged to validate compliance with PCI DSS are: · **Electronic merchants** · **Hotels F2F (Face to face)** · **Airlines**	Quarterly scanning of network vulnerability with an ASV (Approved Scanning Providers)	Specialist from the security company.

The obligations shown in the table are compulsory for all merchants.

Annual audit by QSA

A (QSA) security adviser qualified by the PCI Council carries out an onsite audit, annually, to review the merchant's systems so as to assess the security of the IT systems involved in processing the card data (hardware, software and netware). The merchant is responsible for contracting this audit service from any of the certified audit companies. Banco Sabadell can provide a list of the different specialised audit services.

Quarterly scanning of network vulnerability with an ASV (Approved Scanning Providers)

These are quarterly controls carried out by companies certified by the PCI Council in its capacity as Approved Scanning Vendor (ASV).

Banco Sabadell places at the disposal of its levels 3 and 4 merchants a tool, free of charge, to perform the scanning.

Annual self-evaluation questionnaire (SAQ)

This is an annual questionnaire on the architecture of the merchant's IT system and manner of processing and storing card data.

In accordance with the form and level of access to the card data by the merchant, different SAQ are established (subsets of the standard's requirements).

Banco Sabadell helps its merchants to identify the SAQ that corresponds to them and offers 3 and 4 level merchants an online tool to perform this.

11.7 Banco Sabadell helps you to comply with PCI DSS

To assist with this task, Banco Sabadell offers its customers the **PCI Management Service**, to receive the support of our experts in PCI DSS.

This team will evaluate the type of validation appropriate in accordance with the merchant level and types of access to the company's card data.

You can contact the support service via:

Email:
gestionPCI@bancsabadell.com

Telephone no: 966940426

Banco Sabadell provides its merchants a totally free tool which:

Enables them to **perform their own self-evaluation online questionnaire (SAQ)**.

Perform quarterly vulnerability scannings of PCI (requisite 11.2), if their questionnaire requires it.

Annex I.
Payment form data

DATUM	NAME OF FIELD	COMMENTS
Signature version:	Ds_SignatureVersion	Constant indicating the signature version being used.
Details of the operation	Ds_MerchantParameters	Chain in JSON format with all the parameters of the request encoded in Base 64.
Signature	Ds_Signature	Result of the HMAC SHA256 of the chain ncoded JSON chain in Base 64 sent in the previous parameter.

To create the Ds_MerchantParameters field, all the fields marked as required in the table below must be used.

The remaining fields are optional and may be included if the merchant wishes.

DATUM	NAME OF FIELD	LENGTH	COMMENTS
Merchant number FUC code	Ds_Merchant_ MerchantCode	9 N	**Required**. Fixed code assigned by Banco Sabadell.
Terminal number	Ds_Merchant_ Terminal	3 N	**Required**. Standard: 1 – Operations in euros (Ds_Merchant_Currency= 978) If more terminals are necessary, contact the Banco Sabadell technical service. Terminal number assigned by bank. Three is considered the maximum length.
Order number	Ds_Merchant_ Order	min. 4N max.12 AN For “Card on File” in field must be max. 10 positions. The Virtual POS will add two more positions indicating the payment order number.	**Required**. The first 4 digits must be numerical; the remaining digits can only use the following ASCII characters Del 30 = 0 al 39 = 9 Del 65 = A al 90 = Z Del 97 = a al 122 = z The code must be different from previous transactions.
Amount	Ds_Merchant_ Amount	12 N	**Required**. The last two positions are considered decimals, except in Yen.
Currency	Ds_Merchant_ Currency	4 N	**Required**. 978 - EURO 840 - USD 826 - GBP 392 - JPY 756 - CHF 124 - CAD 4 is considered the maximum length.

Type of transaction	Ds_Merchant_ TransactionType	1 N	**Required**. 3 – Standard payment 4 – Pre-authorisation 5 – Confirmation of pre-authorisation 3 – Partial or full refund 9 – Authentication 10 – Confirmation of authentication 9 – Cancellation of pre-authorisation L – Initial Card on File M – Successive Card on File O – Deferred pre-authorisation P – Confirmation of pre-authorisation Deferred Q – Cancellation of Deferred VPre-authorisation
Product Description	Ds_Merchant_ ProductDescription	Max.125 AN	**Optional**. This field is shown to the holder on the purchase confirmation screen.
Name and surnames of holder	Ds_Merchant_ Titular	Max. 60 AN	This field is shown to the holder on the purchase confirmation screen.
URL	Ds_Merchant_ MerchantURL	250 AN	**Required**. Required if merchant has online notification. URL of merchant which will receive a post with the transaction data.
URLOK	Ds_Merchant_UrlOK	250 AN	**Optional**. If sent it will be used as URLOK, ignoring that configured in the administration module if any.
URLKO	Ds_Merchant_UrlKO	250 AN	**Optional**. If sent it will be used as URLKO, ignoring that configured in the administration module if any.
Name of merchant	Ds_Merchant_ MerchantName	25 AN	**Optional**. Name of merchant appearing on customer payment page, if any.
Holder's language	Ds_Merchant_ ConsumerLanguage	3 N	**Required**. 0 – Customer 1 – Spanish 2 – English 3 – Catalan 4 – French 5 – German 6 – Dutch 7 – Italian 8 – Swedish 9 – Portuguese 10 – Valencian 11 – Polish 12 – Galician 13 – Basque

Merchant data	Ds_Merchant_ MerchantData	1024 AN	**Optional**. Free information of merchant to be received in online response (via URL or e-mail).
Authorisation code	Ds_Merchant_ AuthorisationCode	6 N	**Optional**.
Identifier	Ds_Merchant_ Identifier	Max. 40 AN	**Exclusive field for payment by reference.** Value of the field is Required for first payment transaction. For subsequent payments, the value will be the identifier that the Bank has sent in the first payment response message.
Group of merchants	Ds_Merchant_Group	Max. 9 N	**Exclusive field for payment by reference.** Allows to associate an identifier to a set of merchants.
Additional screens	Ds_Merchant_ DirectPayment	'True' or 'false'	**Exclusive field for payment by reference.** This parameter acts as a flag to indicate if additional screens must be shown

Described below are the fields relating to the card details, given the possibility that they may be captured by the merchant. These new fields will only have to be sent by merchants who capture the data on the card themselves (See the requirements of the PCI-DSS security programme explained in section 11th of this manual).

DATUM	NAME OF FIELD	LENGTH	COMMENTS
Card number	Ds_Merchant_Pan	16-19 N	**Obligatorio** para todas las operaciones, excepto para aquellas que no necesiten de datos de tarjeta, como las confirmaciones o devoluciones.
Expiry date	Ds_Merchant_ ExpiryDate	4 N	**Obligatorio** para todas las operaciones, excepto para aquellas que no necesiten de datos de tarjeta, como las confirmaciones o devoluciones.
CVV2	Ds_Merchant_ CVV2	3 N	**Optional** CVV2/CVC2 code on the card sent*

* The transaction types 2 / 3 / 6 / 8 / 9 / P / Q do not require that the card number, expiry date and CVV2 was informed. In these cases the field (Ds_Merchant_Order) has to be the same that the original transaction.

Annex II.
Error codes

SISxxxx	FIELD AFFECTED	REASON	MESSAGE
SIS0007		Error disassembling input XML	MSG0008
SIS0008	Ds_Merchant_MerchantCode	Field missing	MSG0008
SIS0009	Ds_Merchant_MerchantCode	Format error	MSG0008
SIS0010	Ds_Merchant_Terminal	Field missing	MSG0008
SIS0011	Ds_Merchant_Terminal	Format error	MSG0008
SIS0014	Ds_Merchant_Order	Format error	MSG0008
SIS0015	Ds_Merchant_Currency	Field missing	MSG0008
SIS0016	Ds_Merchant_Currency	Format error	MSG0008
SIS0018	Ds_Merchant_Amount	Field missing	MSG0008
SIS0019	Ds_Merchant_Amount	Format error	MSG0008
SIS0020	Ds_Merchant_Signature	Field missing	MSG0008
SIS0021	Ds_Merchant_Signature	Field empty	MSG0008
SIS0022	Ds_TransactionType	Format error	MSG0008
SIS0023	Ds_TransactionType	Unknown value	MSG0008
SIS0024	Ds_ConsumerLanguage	Value exceeds 3 positions	MSG0008
SIS0025	Ds_ConsumerLanguage	Format error	MSG0008
SIS0026	Ds_Merchant_MerchantCode	Error Merchant inexistent / Terminal sent	MSG0008
SIS0027	Ds_Merchant_Currency	Error currency does not match that assigned for that Terminal.	MSG0008
SIS0028	Ds_Merchant_MerchantCode	Error Merchant/Terminal is de-registered	MSG0008
SIS0030	Ds_TransactionType	In card payment a type of operation has arrived which is not payment nor pre-authorisation	MSG0000
SIS0031	Ds_Merchant_TransactionType	Method of payment not defined	MSG0000
SIS0034		Error accessing database	MSG0000
SIS0038		Error in JAVA	MSG0000
SIS0040		The merchant / Terminal has not assigned method of payment	MSG0008
SIS0041 SIS0042	Ds_Merchant_Signature	Error calculating HASH algorithm	MSG0008
SIS0043		Error making online notification	MSG0008
SIS0046		Card Pin not registered	MSG0002
SIS0051	Ds_Merchant_Order	Repeated order number	MSG0001
SIS0054	Ds_Merchant_Order	No operation to make refund	MSG0008
SIS0055	Ds_Merchant_Order	Operation to be refunded is not valid	MSG0008

SIS0056	Ds_Merchant_Order	Operation to be refunded is not authorised	MSG0008
SIS0057	Ds_Merchant_Amount	Amount to be refunded exceeds limit	MSG0008
SIS0058		Inconsistent data in validation of confirmation	MSG0008
SIS0059	Ds_Merchant_Order	Error, operation for confirmation does not exist	MSG0008
SIS0060	Ds_Merchant_Order	Confirmation for this pre-authorisation already exists	MSG0008
SIS0061	Ds_Merchant_Order	The pre-authorisation to be confirmed is not authorised	MSG0008
SIS0062	Ds_Merchant_Amount	Amount to be confirmed exceeds limit	MSG0008
SIS0063 SIS0064 SIS0065		Error in card number	MSG0008
SIS0066 SIS0067 SIS0068 SIS0069 SIS0070		Error in card expiry date	MSG0008
SIS0071		Expired card	MSG0000
SIS0072	Ds_Merchant_Order	Operation cannot be cancelled	MSG0000
SIS0074	Ds_Merchant_Order	Field missing	MSG0008
SIS0075	Ds_Merchant_Order	Value has fewer than 4 positions or more than 12	MSG0008
SIS0076	Ds_Merchant_Order	Value is not numerical	MSG0008
SIS0078	Ds_TransactionType	Unknown value	MSG0005
SIS0079	Ds_TransactionType	Error when make payment by card	MSG0008
SIS0081	Ds_TransactionType	The session is new, you have lost the data stored	MSG0008
SIS0089	Ds_TransactionType	Ds_Merchant_ExpiryDate value not set in 4 positions	MSG0008
SIS0092	Ds_TransactionType	Ds_Merchant_ExpiryDate value is null	MSG0008
SIS0093		Card not found within table of ranges	MSG0006
SIS0094		Card not authenticated as 3D Secure	MSG0004
SIS0112	Ds_TransactionType	Value not allowed	MSG0008
SIS0114		A GET has been called instead of a POST	MSG0000
SIS0115	Ds_Merchant_Order	No operation to make instalment payment	MSG0008
SIS0116	Ds_Merchant_Order	Operation for instalment payment is not valid.	MSG0008
SIS0117	Ds_Merchant_Order	Operation for instalment payment is not authorised.	MSG0008
SIS0132		The Confirmation of Authorisation date cannot exceed pre-authorisation date by more than 7 days	MSG0008

SIS0133		The confirmation of Authentication date cannot exceed prior authentication by more than 45 days	MSG0008
SIS0139		Initial recurrent payment is duplicated	MSG0008
SIS0142		Time exceeded for payment	MSG0000
SIS0198		Amount exceeds limit allowed for merchant	MSG0008
SIS0199		The number of operations exceeds limit allowed for merchant	MSG0008
SIS0200		Amount accumulated exceeds limit allowed for merchant	MSG0008
SIS0214		Merchant does not accept refunds	MSG0008
SIS0216		The CVV2 has more than three positions	MSG0008
SIS0217		Format error in CVV2	MSG0008
SIS0218		"Operations" input does not allow secure payments	MSG0008
SIS0219		The number of card operations exceeds limit allowed for merchant	MSG0008
SIS0220		Accumulated amount of card exceeds limit allowed for merchant	MSG0008
SIS0221		Error. The CVV2 is required:	MSG0008
SIS0222		Cancellation for this pre-authorisation already exists	MSG0008
SIS0223		The pre-authorisation to be cancelled is not authorised	MSG0008
SIS0224		Merchant does not allow cancellations due to lack of extended signature	MSG0008
SIS0225		No operation to make cancellation	MSG0008
SIS0226		Inconsistent data in validation of a cancellation	MSG0008
SIS0227	Ds_Merchant_Transac-tionDate	Invalid value	MSG0008
SIS0229		No deferred payment code requested	MSG0008
SIS0252		Merchant does not allow card to be sent	MSG0008
SIS0253		Card does not comply with check-digit	MSG0008
SIS0254		The number of operations per IP exceeds limit allowed for merchant	MSG0008
SIS0255		Amount accumulated per IP exceeds limit allowed for merchant	MSG0008
SIS0256		Merchant cannot perform pre-authorisations.	MSG0008
SIS0257		Card does not allow pre-authorisations	MSG0008
SIS0258		Inconsistent confirmation data	MSG0008

SIS0261		Operation exceeds an operating limit defined by Banco Sabadell	MSG0008
SIS0270	Ds_Merchant_Transac-tionType	Type of operation not activated for this merchant	MSG0008
SIS0274	Ds_Merchant_Transac-tionType	Type of operation unknown or not allowed for this input to the Virtual POS.	MSG0008
SIS0281		Operation exceeds an operating limit defined by Banco Sabadell	MSG0008
SIS0296		Error validating initial operation data “Card on File (Subscriptions P./Express P.)”.	MSG0008
SIS0297		Maximum number of operations exceeded (99 oper. or 1 year) for successive transactions in “Card on File (Subscriptions P. /Express P.)”.A new “Initial File Card” operation is necessary to start the cycle.	MSG0008
SIS0298		The merchant does not allow Card on File operations	MSG0008
SIS0319		The merchant does not belong to the group specified in Ds_Merchant_Group	MSG0008
SIS0321		The reference indicated in Ds_Merchant_ Identifier is not associated with the merchant	MSG0008
SIS0322		Format error in Ds_Merchant_Group	MSG0008
SIS0325		It was requested not to show screens but no card reference has been sent	MSG0008
SIS0448		An operation was performed with a DINERS card but the merchant does not have this type of card enabled. To enable it they must contact Diners Club directly.	MSG0008
SIS0449		An “A” type transaction has been sent and the merchant does not have the operation activated for this type of transaction.	MSG0008
SIS0450		An “A” type transaction has been sent with an American Express card and the merchant does not have the operation activated for this type of transaction.	MSG0008
SIS0451		An “A” type transaction has been sent and the merchant does not have the operation activated for this type of transaction.	MSG0008
SIS0452		A 4B card has been used and the merchant does not accept this type of card.	MSG0008
SIS0453		A JCB card has been used and the merchant does not accept this type of card.	MSG0008
SIS0454		An American Express card has been used and the merchant does not accept this type of card. To enable it they must contact American Express directly.	MSG0008
SIS0455		Method of payment not available	MSG0008
SIS0456		Payment method unsecure (Visa) not available	MSG0008

SIS0457		A business card has been used and the merchant does not accept this type of card. To enable it it must contact the managing office.	MSG0008
SIS0458		A business card has been used and the merchant does not accept this type of card. To enable it it must contact the managing office.	MSG0008
SIS0459		A JCB card has been used and the merchant does not accept this type of card.	MSG0008
SIS0460		An American Express card has been used and the merchant does not accept this type of card.	MSG0008
SIS0461		An American Express card has been used and the merchant does not accept this type of card.	MSG0008
SIS0462		Error, a secure request has been sent Host to Host.	MSG0008
SIS0463		Method of payment not available	MSG0008
SIS0464		A business card has been used and the merchant does not accept this type of card. To enable it it must contact the managing office.	MSG0008
SIS0465		An unsecure payment request was launched and the merchant does not accept non-secure payments.	MSG0008

The table below lists the messages the payment page may show the cardholder for the various errors which may occur.

CODE	MESSAGE
MSG0000	System occupied, try later
MSG0001	Repeated order number
MSG0002	Card Pin not registered on FINANET
MSG0003	System launching, try again in a few moments
MSG0004	Authentication Error
MSG0005	No valid payment method exists for your card
MSG0006	Non-service card
MSG0007	Data missing, please check your browser accepts cookies
MSG0008	Error in data sent. Contact your merchant.

Annex III.
Table of response codes (ds_response)

A. CODES FOR APPROVED TRANSACTIONS

CODE	TITLE	DESCRIPTION
000	TRANSACTION APPROVED	Transaction authorised by card issuing bank
001	TRANSACTION APPROVED AFTER IDENTIFICATION OF HOLDER	Exclusive code for transactions Verified by Visa or MasterCard SecureCode. The transaction has been authorised and the issuing bank informs us that it has correctly authenticated the identity of the cardholder.
002 - 099	TRANSACTION APPROVED	Transaction authorised by issuing bank

B. CODES FOR TRANSACTIONS REJECTED

B.1 Transactions rejected due to general motives

CODE	TITLE	DESCRIPTION
101	EXPIRED CARD	Transaction rejected because card expiry date entered during payment is prior to that currently valid.
102	CARD TEMPORARILY BLOCKED OR UNDER SUSPICION OF FRAUD	Card temporarily blocked by issuing bank or under suspicion of fraud
104	OPERATION NOT ALLOWED	Operation not allowed for this type of card.
106	NO. ATTEMPTS EXCEEDED	Number of attempts with erroneous PIN exceeded.
107	CONTACT ISSUER	Issuing bank does not allow automatic authorisation. It is necessary to call your authorisation centre to obtain manual approval.
109	IDENTIFICATION OF MERCHANT OR TERMINAL INVALID	Rejected because merchant is not correctly registered in international card systems.
110	AMOUNT INVALID	Transaction amount unusual for this type of merchant requesting payment authorisation.
114	CARD DOES NOT SUPPORT TYPE OF OPERATION REQUESTED	Operation not allowed for this type of card.
116	INSUFFICIENT BALANCE	The cardholder has insufficient credit to meet payment.
118	CARD NOT REGISTERED	Card inexistent or not registered by issuing bank.
119	TRANSACTION REJECTED	Transaction rejected Contact the Bank.
125	CARD NOT EFFECTIVE	Card inexistent or not registered by issuing bank.

129	CVV2/CVC2 ERROR.	The CVV2/CVC2 code (three digits on back of card) entered by consumer is erroneous.
167	CONTACT ISSUER SUSPECTED FRAUD	Due to suspicion that transaction is fraudulent the issuing bank does not allow automatic authorisation. It is necessary to call your authorisation centre to obtain manual approval.
180	NON-SERVICE CARD	Operation not allowed for this type of card.
181-182	CARD WITH DEBIT OR CREDIT RESTRICTIONS	Card temporarily blocked by issuing bank
184	AUTHENTICATION ERROR	Exclusive code for transactions Verified by Visa or MasterCard SecureCode. Transaction rejected because issuing bank cannot authenticate the cardholder.
190	REJECTION WITHOUT SPECIFYING MOTIVE	Transaction rejected by issuing bank but without reporting the reason.
191	ERRONEOUS EXPIRY DATE	Transaction rejected because card expiry date entered during payment does not match that currently valid.

B.2 Transactions rejected due to motives in which the card issuing bank considers there are signs of fraud.

CODE	TITLE	DESCRIPTION
201	EXPIRED CARD	Transaction rejected because card expiry date entered during payment is prior to that currently valid. In addition, the issuing bank considers that the card is subject to possible fraud.
202	CARD TEMPORARILY BLOCKED OR UNDER SUSPICION OF FRAUD	Card temporarily blocked by issuing bank or under suspicion of fraud In addition, the issuing bank considers that the card is subject to possible fraud.
204	OPERATION NOT ALLOWED	Operation not allowed for this type of card. In addition, the issuing bank considers that the card is subject to possible fraud.
207	CONTACT ISSUER	Issuing bank does not allow automatic authorisation. It is necessary to call your authorisation centre to obtain manual approval. In addition, the issuing bank considers that the card is subject to possible fraud.
208 - 209	CARD LOST OR STOLEN	Card blocked by issuing bank as holder has reported it is stolen or lost. In addition, the issuing bank considers that the card is subject to possible fraud.

280	CVV2/CVC2 ERROR.	Exclusive code for transactions in which 3-figit CVV2 code is requested (Visa card) or CVC2 (MasterCard) on back of card. The CVV2/CVC2 code entered by purchaser is erroneous. In addition, the issuing bank considers that the card is subject to possible fraud.
290	REJECTION WITHOUT SPECIFYING MOTIVE	Transaction rejected by issuing bank but without reporting the reason. In addition, the issuing bank considers that the card is subject to possible fraud.

C. CÓDIGOS REFERIDOS A ANULACIONES O DEVOLUCIONES (Ds_Merchant_TransactionType = 3) SOLICITADAS POR EL COMERCIO)

CODE	TITLE	DESCRIPTION
400	CANCELLATION ACCEPTED	Cancellation or partial chargeback transaction accepted by issuing bank.
480	ORIGINAL OPERATION NOT FOUND OR TIMED OUT	The cancellation or partial chargeback not accepted because original operation not located or because issuing bank has not responded within predefined time-out limit.
481	CANCELLATION ACCEPTED	Cancellation or partial chargeback transaction accepted by issuing bank. However, issuing bank response received late, outside predefined time-out limit.

D. CÓDIGOS REFERIDOS A CONCILIACIONES DE PRE-AUTORIZACIONES O PRE-AUTENTICACIONES (Ds_Merchant_TransactionType = 2, 8, O o R)

CODE	TITLE	DESCRIPTION
500	RECONCILIATION ACCEPTED	Reconciliation transaction accepted by issuing bank.
501 - 503	ORIGINAL OPERATION NOT FOUND OR TIME-OUT EXCEEDED	The reconciliation was not accepted because original operation not located or because issuing bank has not responded within predefined time-out limit.

9928	CANCELLATION OF PRE—AUTHORISATION PERFORMED BY SYSTEM	System has cancelled deferred pre-authorisation as over 72 hours have passed.
9929	CANCELLATION OF PRE-AUTHORISATION PERFORMED BY MERCHANT	The cancellation of the pre-authorisation was accepted

E. ERROR CODES SENT BY PAYMENT GATEWAY OF BANCO SABADELL

CODE	TITLE	DESCRIPTION
904	MERCHANT NOT REGISTERED IN FUC	There is a problem in configuration of merchant code. Contact Banco Sabadell to solve it.
909	SYSTEM ERROR	Error in stability of Banco Sabadell payment gateway or exchange systems of Visa or MasterCard.
912	ISSUER NOT AVAILABLE	Authorising centre of issuing bank not operational at this time.
913	DUPLICATED TRANSMISSION	A transaction with the same order number was recently processed (Ds_Merchant_Order).
916	AMOUNT TOO SMALL	Not possible to operate with this amount.
928	TIME-OUT EXCEEDED	Issuing bank does not respond to authorisation request within predefined time-out.
940	TRANSACTION CANCELLED EARLIER	Cancellation or partial chargeback of a transaction requested which was already cancelled.
941	AUTHORISATION TRANSACTION ALREADY CANCELLED BY PREVIOUS CANCELLATION	Confirmation of a transaction is being requested with an order number (Ds_Merchant_Order) which matches an operation already cancelled.
942	ORIGINAL AUTHORISATION TRANSACTION REJECTED	Confirmation of a transaction is being requested with an order number (Ds_Merchant_Order) which matches an operation already rejected.
943	DIFFERENT ORIGINAL TRANSACTION DATA	An erroneous confirmation is being requested.
944	ERRONEOUS SESSION	A third session is being requested. In the payment process only two sessions may be open (the current one and previous pending closure).
945	DUPLICATED TRANSMISSION	A transaction with the same order number was recently processed (Ds_Merchant_Order).
946	OPERATION TO BE CANCELLED IN PROGRESS	Cancellation or partial chargeback of an original transaction is requested which is still in progress and pending response.

947	DUPLICATED TRANSMISSION IN PROGRESS	A transaction with the same order number is being attempted (Ds_Merchant_Order) of another still pending response.
949	TERMINAL NON-OPERATIONAL	The merchant number (Ds_Merchant_MerchantCode) or terminal (Ds_Merchant_Terminal) are not registered or not operational.
950	REFUND NOT ALLOWED	Refund not allowed by regulation.
965	COMPLIANCE INFRINGEMENT	Infringement of Visa or Mastercard compliance
9064	CARD LENGTH INCORRECT	No. positions of card incorrect
9078	NO PAYMENT METHOD EXISTS	The types of payment defined for the terminal (Ds_ Merchant_Terminal) by the transaction processor do not allow payment with the type of card entered.
9093	CARD DOES NOT EXIST:	Inexistent card
9094	REJECTION OF ISSUERS	Operation rejected by international issuers
9104	SECURE OPER. NOT POSSIBLE	Merchant with obligatory authentication and holder without secure purchase code
9126	OPERATION REJECTED TO AVOID DUPLICITIES	
9142	PAYMENT TIME LIMIT EXCEEDED	The cardholder not authenticated during maximum time allowed.
9218	SECURE OPERATIONS CANNOT BE PERFORMED	The Operations input does not allow Secure operations
9253	CHECK-DIGIT ERRONEOUS	Card does not comply with check-digit (position 16 of card number calculated using Luhn algorithm).
9256	PRE-AUTHORISATIONS NOT ENABLED	Card cannot perform Pre-authorisations
9261	OPERATING LIMIT EXCEEDED	Transaction exceeds operating limit set by Banco Sabadell
9280	EXCEEDS BLOCKING ALERTS	The operation exceeds the blocking alerts; cannot be processed
9281	EXCEEDS BLOCKING ALERTS	The operation exceeds the blocking alerts; cannot be processed
9283	SUPERA ALERTAS BLOQUEANTES	La operación excede las alertas bloqueantes, no se puede procesar.
9334	REJECTION DUE TO SECURITY FILTERS	The alert was blocked by the security filters

9912	ISSUER NOT AVAILABLE	Authorising centre of issuing bank not operational at this time.
9913	ERROR IN CONFIRMATION	Error in the confirmation sent by merchant to Virtual POS (only applicable in SOAP synchronisation option)
9914	CONFIRM “KO”	Confirmation “KO” of merchant (only applicable in SOAP synchronisation option)
9915	PAYMENT CANCELLED	User has cancelled payment
9928	DEFERRED AUTHORISATION CANCELLED	Cancellation of deferred authorisation made by SIS (batch process)
9929	DEFERRED AUTHORISATION CANCELLED	Cancellation of deferred authorisation made by merchant
9997	SIMULTANEOUS TRANSACTION	The Virtual POS is simultaneously processing another operation with the same card.
9998	OPERATION STATUS REQUESTED	Temporary status while operation is processed. When the operation ends this code will change.
9999	OPERATION STATUS AUTHENTICATING	Temporary status while POS authenticates holder. Once this process has finalised, the POS will assign a new code to the operation.

Annex IV.
ISO Country codes

Annex IV. ISO country codes.

004	Afghanistan
008	Albania
012	Algeria
016	American Samoa
020	Andorra
024	Angola
028	Antigua and Barbuda
031	Azerbaijan
032	Argentina
036	Australia
040	Austria
044	Bahamas
048	Bahrain
050	Bangladesh
051	Armenia
052	Barbados
056	Belgium
060	Bermuda
064	Bhutan
068	Bolivia, Plurinational State of
070	Bosnia and Herzegovina
072	Botswana
074	Bouvet Island
076	Brazil
084	Belize
086	British Indian Ocean Territory
090	Solomon Islands
092	Virgin Islands, British
096	Brunei Darussalam
100	Bulgaria
104	Myanmar
108	Burundi
112	Belarus
116	Cambodia
120	Cameroon
124	Canada
132	Cape Verde
136	Cayman Islands
140	Central African Republic
144	Sri Lanka
148	Chad
152	Chile
156	China
158	Taiwan, Province of China
162	Christmas Island
166	Cocos (Keeling) Islands
170	Colombia
174	Comoros
175	Mayotte
178	Congo
180	Congo, the Dem. Rep. of the
184	Cook Islands
188	Costa Rica
191	Croatia
192	Cuba
196	Cyprus
203	Czech Republic
204	Benin
208	Denmark
212	Dominica
214	Dominican Republic
218	Ecuador
222	El Salvador
226	Equatorial Guinea
231	Ethiopia
232	Eritrea
233	Estonia
234	Faroe Islands
238	Falkland Islands (Malvinas)
239	S. Georgia and the S. Sandwich Isl.
242	Fiji
246	Finland
248	Aland Islands
250	France
254	French Guiana
258	French Polynesia
260	French Southern Territories
262	Djibouti
266	Gabon
268	Georgia
270	Gambia
275	Palestinian Territory, Occupied
276	Germany
288	Ghana
292	Gibraltar
296	Kiribati
300	Greece
304	Greenland
308	Grenada
312	Guadeloupe
316	Guam
320	Guatemala
324	Guinea
328	Guyana
332	Haiti
334	Heard Isl. and McDonald Isl.
336	Holy See (Vatican City State)
340	Honduras
344	Hong Kong
348	Hungary
352	Iceland
356	India
360	Indonesia
364	Iran, Islamic Rep. of
368	Iraq
372	Ireland
376	Israel
380	Italy
384	Côte d'Ivoire
388	Jamaica
392	Japan
398	Kazakhstan
400	Jordan
404	Kenya
408	Korea, Dem. People's Rep. of
410	Korea, Republic of
414	Kuwait
417	Kyrgyzstan
418	Lao People's Dem. Rep.
422	Lebanon
426	Lesotho
428	Latvia
430	Liberia

Annex IV. ISO country codes.

434	Libya
438	Liechtenstein
440	Lithuaniaa
442	Luxembourg
446	Macao
450	Madagascar
454	Malawi
458	Malaysia
462	Maldives
466	Mali
470	Malta
474	Martinique
478	Mauritania
480	Mauritius
484	Mexico
492	Monaco
496	Mongolia
498	Moldova, Republic of
499	Montenegro
500	Montserrat
504	Morocco
508	Mozambique
512	Oman
516	Namibia
520	Nauru
524	Nepal
528	Netherlands
531	Curaçao
533	Aruba
540	New Caledonia
548	Vanuatu
554	New Zealand
558	Nicaragua
562	Niger
566	Nigeria
570	Niue
574	Norfolk Island
578	Norway
580	Northern Mariana Islands
583	Micronesia, Fed. States of
584	Marshall Islands
585	Palau
586	Pakistan
591	Panama
598	Papua New Guinea
600	Paraguay
604	Peru
608	Philippines
612	Pitcairn
616	Poland
620	Portugal
624	Guinea-Bissau
626	Timor-Leste
630	Puerto Rico
634	Qatar
638	Reunion
642	Romania
643	Russian Federation
646	Rwanda
652	Saint Barthélemy
654	St. Helena, Ascension & T. da Cunha
659	Saint Kitts and Nevis
660	Anguilla
662	Saint Lucia
663	Saint Martin (French part)
666	Saint Pierre and Miquelon
670	Saint Vincent and the Grenadines
674	San Marino
678	Sao Tome and Principe
682	Saudi Arabia
686	Senegal
688	Serbia
690	Seychelles
694	Sierra Leone
702	Singapore
703	Slovakia
704	Viet Nam
705	Slovenia
706	Somalia
710	South Africa
716	Zimbabwe
724	Spain
728	South Sudan
729	Sudan
732	Western Sahara
740	Suriname
744	Svalbard and Jan Mayen
748	Swaziland
752	Sweden
756	Switzerland
760	Syrian Arab Republic
762	Tajikistan
764	Thailand
768	Togo
772	Tokelau
776	Tonga
780	Trinidad and Tobago
784	United Arab Emirates
788	Tunisia
792	Turkey
795	Turkmenistan
796	Turks and Caicos Islands
798	Tuvalu
800	Uganda
804	Ukraine
807	Macedonia
818	Egypt
826	United Kingdom
831	Guernsey
832	Jersey
833	Isle of Man
834	Tanzania, United Republic of
840	United States
850	Virgin Islands, U.S.
854	Burkina Faso
858	Uruguay
860	Uzbekistan
862	Venezuela, Bolivarian Rep. of
876	Wallis and Futuna
882	Samoa
887	Yemen
894	Zambia

Annex V.
ISO Currency Codes

Annex V. ISO currency codes.

Lek	ALL	8
Algerian Dinar	DZD	12
Angola Kwanza	AON	24
Argentine Peso	ARS	32
Australian Dollar	AUD	36
Bahamian Dollar	BSD	44
Bahraini Dinar	BHD	48
Taka	BDT	50
Armenian Dram	AMD	51
Barbados Dollar	BBD	52
Bermudian Dollar	BMD	60
Ngultrum	BTN	64
Boliviano	BOB	68
Dinar	BAM	70
Pula	BWP	72
Cruzeiro	BRC	76
Belize Dollar	BZD	84
Solomon Islands Dollar	SBD	90
Brunei Dollar	BND	96
Kyat	MMK	104
Burundi Franc	BIF	108
Bellarusian Ruble	BYB	112
Riel	KHR	116
Canadian Dollar	CAD	124
Cape Verde Escudo	CVE	132
Cayman Islands Dollar	KYD	136
Sri Lanka Rupee	LKR	144
Chilean Peso	CLP	152
Yuan Renminbi	CNY	156
Chinese Renmimbi	CNH	157
Chinese Renmimbi	CNX	158
Colombian Peso	COP	170
Comoro Franc	KMF	174
Costa Rican Colon	CRC	188
Croatian Kuna	HRK	191
Cuban Peso	CUP	192
Cyprus Pound	CYP	196
Koruna	CSK	200
Czech Koruna	CZK	203
Danish Krone	DKK	208
Dominican Peso	DOP	214
El Salvador Colon	SVC	222
Ethiopian Birr	ETB	230
Nakfa	ERN	232
Kroon	EEK	233
Falkland Islands Pound	FKP	238
Fiji Dollar	FJD	242
Djibouti Franc	DJF	262
Dalasi	GMD	270
Ghana Cedi	GHC	288
Gibraltar Pound	GIP	292
Quetzal	GTQ	320
Guinea Franc	GNF	324
Guyana Dollar	GYD	328
Gourde	HTG	332
Lempira	HNL	340
Hong Kong Dollar	HKD	344
Forint	HUF	348
Iceland Krona	ISK	352
Indian Rupee	INR	356
Rupiah	IDR	360
Iraqi Dinar	IQD	368
New Israeli Sheqel	ILS	376
Jamaican Dollar	JMD	388
Yen	JPY	392
Tenge	KZT	398
Jordanian Dinar	JOD	400
Kenyan Shilling	KES	404
Won	KRW	410
Kuwaiti Dinar	KWD	414
Som	KGS	417
Kip	LAK	418
Lebanese Pound	LBP	422
Loti	LSL	426
Latvian Lats	LVL	428
Liberian Dollar	LRD	430
Libyan Dinar	LYD	434
Lithuanian Litas	LTL	440
Pataca	MOP	446
Malagassy Franc	MGF	450
Kwacha	MWK	454
Malaysian Ringgit	MYR	458
Rufiyaa	MVR	462
Mali	MLF	466
Maltese Lira	MTL	470
Ouguiya	MRO	478
Mauritius Rupee	MUR	480
Mexican Peso	MXN	484
Tugrik	MNT	496
Moldovan Leu	MDL	498
Moroccan Dirham	MAD	504
Rial Omani	OMR	512

Annex V. ISO currency codes.

Namibia Dollar	NAD	516
Nepalese Rupee	NPR	524
Netherlands Antillian Guilder	ANG	532
Aruban Guilder	AWG	533
Yugoslavian New Dian	NTZ	536
Vatu	VUV	548
New Zealand Dollar	NZD	554
Naira	NGN	556
Cordoba Oro	NIO	558
Naira	NGN	566
Norwegian Krone	NOK	578
Pacific Island	PCI	582
Pakistan Rupee	PKR	586
Balboa	PAB	590
Kina	PGK	598
Guarani	PYG	600
Nuevo Sol	PEN	604
Philippine Peso	PHP	608
Guinea-Bissau Peso	GWP	624
Timor Escudo	TPE	626
Qatari Rial	QAR	634
Russian Ruble	RUB	643
Rowanda Franc	RWF	646
Saint Helena Pound	SHP	654
Dobra	STD	678
Saudi Riyal	SAR	682
Seychelles Rupee	SCR	690
Leone	SLL	694
Singapore Dollar	SGD	702
Dong	VND	704
Slovenian Tolar	SIT	705
Somali Shilling	SOS	706
Rand	ZAR	710
Zimbabwe Dollar	ZWD	716
Yemeni Dinar	YDD	720
Sudanese Pound	SDP	736
Sudan Airlines	SDA	737
Lilangeni	SZL	748
Swedish Krona	SEK	752
Swiss Franc	CHF	756
Syrian Pound	SYP	760
Tajik Ruble	TJR	762
Baht	THB	764
Pa'anga	TOP	776
Trinidad and Tobago Dollar	TTD	780
UAE Dirham	AED	784
Tunisian Dinar	TND	788
Turkish Lira	PTL	793
Manat	TMM	795
Uganda Shilling	UGX	800
Karbovanet	UAK	804
Denar	MKD	807
Egyptian Pound	EGP	818
Pound Sterling	GBP	826
Tanzanian Shilling	TZS	834
US Dollar	USD	840
Peso Uruguayo	UYU	858
Uzbekistan Sum	UZS	860
Tala	WST	882
Yemeni Rial	YER	886
Serbian Dinar	CSD	891
Zambian Kwacha	ZMK	894
New Taiwan Dollar	TWD	901
Manat	TMT	934
Cedi	GHS	936
Bolivar Fuerte	VEF	937
Serbian Dinar	RSD	941
Metical	MZN	943
Azerbaijanian Manat	AZN	944
New Leu	RON	946
Turkish Lira	TRY	949
CFA Franc BEAC	XAF	950
East Caribbean Dollar	XCD	951
CFA Franc BCEAO	XOF	952
CFP Franc	XPF	953
European Currency UN	XEU	954
Kwacha	ZMW	967
Surinam Dollar	SRD	968
Malagasy Ariary	MGA	969
Afghani	AFN	971
Somoni	TJS	972
Kwanza	AOA	973
Belarussian Ruble	BYR	974
Bulgarian Lev	BGN	975
Congolese Franc	CDF	976
Convertible Marks	BAM	977
Euro	EUR	978
Hryvnia	UAH	980
Lari	GEL	981
Zloty	PLN	985
Brazilian Real	BRL	986
Peseta Convertible	ESB	995

Annex VI.
Specific examples using Payment of Subscriptions / Express Payments

The annex provides specific examples using the Payment of Subscriptions / Express Payments functionality, for each of modes of processing transactions through the Banco Sabadell Virtual POS.

Below are several examples based on a merchant using the "**realizarPago**" entry point to the Virtual POS.

Example 1:
Payment with request for new identifier.

A payment is made and the Ds_Merchant_ Identifier parameter is added with the value REQUIRED to generate a new identifier which is return in the notification together with the expiry date. The identifier will be associated with the merchant indicated by the Ds_Merchant_MerchantCode parameter.

Data to be sent in the request:

Ds_Merchant_MerchantCode=327234688
Ds_Merchant_Terminal=1
Ds_Merchant_Currency=978
Ds_Merchant_TransactionType=0
Ds_Merchant_Amount=100
Ds_Merchant_Order=112545
Ds_Merchant_Identifier=REQUIRED
Ds_Merchant_MerchantURL=<URL de notificación>

Response in on-line notification.

If the authorisation is authorised, the On-Line notification and URL OK will include the new parameters with the value of the identifier generated and the expiry date of the card by way of identifier expiry date. The other fields of the notification do not differ and neither does the signature calculation.

An example of an identifier would be as follows:

Ds_Merchant_Identifier=a091f0f9f0aaf0506930dda4a6974f1d-f4a0d9c1
Ds_ExpiryDate=2012

Example 2:
Payment with an identifier.

A payment is made and the Ds_Merchant_ Identifier parameter is added with the identifier value to be used. In this case, the Ds_Merchant_DirectPayment parameter could be used with the value 'true' and the performance would be the same.

Data to be sent in the request:

Ds_Merchant_MerchantCode=327234688
Ds_Merchant_Terminal=1
Ds_Merchant_Currency=978
Ds_Merchant_TransactionType=0
Ds_Merchant_Amount=100
Ds_Merchant_Order=112546<No debe ser el mismo de la operación original>
Ds_Merchant_Identifier=a091f0f9f0aaf0506930dda4a6974f1d-f4a0d9c1
Ds_Merchant_MerchantURL=<URL de notificación>

Example 3:
Payment with request for new identifier and group code.

A payment is made and the Ds_Merchant_ Identifier parameter is added with the value REQUIRED to generate a new identifier which is return in the notification together with the expiry date. The identifier will be associated with the merchant group indicated in the Ds_Merchant_Group field.

Data to be sent in the request:

Ds_Merchant_MerchantCode=327234688
Ds_Merchant_Terminal=1
Ds_Merchant_Currency=978
Ds_Merchant_TransactionType=0
Ds_Merchant_Amount=100
Ds_Merchant_Order=112545
Ds_Merchant_Identifier=REQUIRED
Ds_Merchant_Group=777888991
Ds_Merchant_MerchantURL=<URL de notificación>

Response in on-line notification

If the authorisation is authorised, the On-Line notification and URL OK will include the new parameters with the value of the identifier

generated and the expiry date of the card by way of identifier expiry date. The other fields of the notification do not differ and neither does the signature calculation.

An example of an identifier would be as follows:

Ds_Merchant_
Identifier=a091f0f9f0aaf0506930dda4a6974f1df4a0d9c1
Ds_ExpiryDate=2012

Example 4:
Payment with an identifier associated with a group.

A payment is made and the Ds_Merchant_ Identifier parameter is added with the identifier value to be used and the Ds_Merchant_ Group field with the group ID. In this case, the Ds_Merchant_DirectPayment parameter could be used with the value 'true' and the performance would be the same.

Data to be sent in the request:

Ds_Merchant_MerchantCode=327234688
Ds_Merchant_Terminal=1
Ds_Merchant_Currency=978
Ds_Merchant_TransactionType=0
Ds_Merchant_Amount=100
Ds_Merchant_Order=112546
Ds_Merchant_
Identifier=a091f0f9f0aaf0506930dda4a6974f1df4a0d9c1
Ds_Merchant_Group=777888991
Ds_Merchant_MerchantURL=<URL de notificación>

Example 5:
Payment with an identifier and flag not to display screens.

A payment is made and the Ds_Merchant_ Identifier parameters are added with the reference value to be used and Ds_Merchant_DirectPayment with the value "true".

Data to be sent in the request:

Ds_Merchant_MerchantCode=327234688
Ds_Merchant_Terminal=1
Ds_Merchant_Currency=978
Ds_Merchant_TransactionType=0
Ds_Merchant_Amount=100
Ds_Merchant_Order=112546 <No tiene que ser el mismo de la operación original>
Ds_Merchant_
Identifier=a091f0f9f0aaf0506930dda4a6974f1df4a0d9c1
Ds_Merchant_MerchantURL=<URL de notificación>
Ds_Merchant_DirectPayment=true

Example 6:
Payment with request for new identifier via Web Service.

A payment is made and the <DS_MERCHANT_IDENTIFIER XML> element is added with the value REQUIRED to generate a new identifier which is returned in the XML response together with the expiry date. The identifier will be associated with the merchant group indicated in the <DS_MERCHANT_GROUP> element if defined. The <DS_MERCHANT_DIRECTPAYMENT> element must not be sent in the original request.

Request XML:

```
<REQUEST>
    <DATOSENTRADA>
    <DS_MERCHANT_AMOUNT>100</DS_MERCHANT_
    AMOUNT>
    <DS_MERCHANT_ORDER>1473846837</DS_MER-
    CHANT_ORDER>
    <DS_MERCHANT_MERCHANTCODE>327234688</
    DS_MERCHANT_MERCHANTCODE>
    <DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CUR-
    RENCY>
    <DS_MERCHANT_PAN>4548812049400004</DS_MER-
    CHANT_PAN>
    <DS_MERCHANT_EXPIRYDATE>2012</DS_MERCHANT_EX-
    PIRYDATE>
    <DS_MERCHANT_CVV2>533</DS_MERCHANT_CVV2>
    <DS_MERCHANT_TRANSACTIONTYPE>A</DS_MERCHANT_
    TRANSACTIONTYPE>
    <DS_MERCHANT_TERMINAL>2</DS_MERCHANT_TER-
    MINAL>
    <DS_MERCHANT_IDENTIFIER>REQUIRED</DS_MER-
    CHANT_IDENTIFIER>
    </DATOSENTRADA>
    <DS_SIGNATUREVERSION>HMAC_SHA256_V1</DS_SIG-
    NATUREVERSION>
    <DS_SIGNATURE>0HMBHIDZY/LZHS5YJTMTUTSUQWSG-
    WOOBOPW5BPSFI5E=</DS_SIGNATURE>
</REQUEST><
```

XML Response

If the authorisation is authorised, the XML response will include the new parameters with the value of the identifier generated and the expiry date of the card by way of identifier expiry date. These data will also be sent in the Online notification. The other fields of the response do not differ and neither does the signature calculation.

An example of the XML response would be as follows:

```
<RETORNOXML>
    <CODIGO>0</CODIGO>
    <OPERACION>
    <DS_AMOUNT>100</DS_AMOUNT>
    <DS_CURRENCY>978</DS_CURRENCY>
    <DS_ORDER>1473846837</DS_ORDER>
    <DS_SIGNATURE>HLKRCQPRW5DE7EFKRULD9QXLS5K7PLV-
    LX2CFWNNCQO4=</DS_SIGNATURE>
    <DS_MERCHANTCODE>327234688</DS_MERCHANT-
    CODE>
    <DS_TERMINAL>2</DS_TERMINAL>
    <DS_RESPONSE>0000</DS_RESPONSE>
    <DS_AUTHORISATIONCODE>229360</DS_AUTHORISATION-
    CODE>
    <DS_TRANSACTIONTYPE>A</DS_TRANSACTIONTYPE>
    <DS_SECUREPAYMENT>0</DS_SECUREPAYMENT>
    <DS_LANGUAGE>1</DS_LANGUAGE>
    <DS_EXPIRYDATE>2012</DS_EXPIRYDATE>
    <DS_MERCHANT_IDENTIFIER>DE021281B7303F-
    3C3B2083A2BB150C21E6574946</DS_MERCHANT_IDEN-
    TIFIER>
    <DS_MERCHANTDATA></DS_MERCHANTDATA>
    <DS_CARD_COUNTRY>724</DS_CARD_COUNTRY>
    </OPERACION>
</RETORNOXML>
```

Example 7: Payment using an identifier via Web Service entry.

A payment is made and the <DS_MERCHANT_IDENTIFIER XML> element is added with the identifier value to be used. In this case, it would be possible to use the <DS_MERCHANT_DIRECTPAYMENT> XML element with the value ‘true’ and the DCC would not apply (if any).

Request XML:

```
<REQUEST>
    <DATOSENTRADA>
    <DS_MERCHANT_AMOUNT>100</DS_MERCHANT_
    AMOUNT>
    <DS_MERCHANT_ORDER>1473847697</DS_MERCHANT_
    ORDER>
    <DS_MERCHANT_MERCHANTCODE>327234688</DS_MER-
    CHANT_MERCHANTCODE>
    <DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CUR-
    RENCY>
    <DS_MERCHANT_PAN>4548812049400004</DS_MER-
    CHANT_PAN>
    <DS_MERCHANT_EXPIRYDATE>2012</DS_MERCHANT_EX-
    PIRYDATE>
    <DS_MERCHANT_CVV2>533</DS_MERCHANT_CVV2>
    <DS_MERCHANT_TRANSACTIONTYPE>A</DS_MERCHANT_
    TRANSACTIONTYPE>
    <DS_MERCHANT_TERMINAL>2</DS_MERCHANT_TERMI-
    NAL>
    <DS_MERCHANT_IDENTIFIER>DE021281B7303F-
    3C3B2083A2BB150C21E6574946</DS_MERCHANT_IDEN-
    TIFIER>
    </DATOSENTRADA>
    <DS_SIGNATUREVERSION>HMAC_SHA256_V1</DS_SIGNA-
    TUREVERSION>
    <DS_SIGNATURE>GKPI63NIDOE7NKK+NVVLUVQYW-
    WDHYQBWMVCQKOHTV6I=</DS_SIGNATURE>
</REQUEST>
```

XML Response

If the operation is authorised, the response XML will also include the parameter with the value of the identifier sent. These data will also be sent in the Online notification. The other fields of the response do not differ and neither does the signature calculation.

An example of the XML response would be as follows:

```
<REQUEST>
    <DATOSENTRADA
    ><DS_MERCHANT_AMOUNT>100</DS_MERCHANT_
    AMOUNT>
    <DS_MERCHANT_ORDER>1473847697</DS_MERCHANT_
    ORDER>
    <DS_MERCHANT_MERCHANTCODE>327234688</DS_MER-
    CHANT_MERCHANTCODE>
    <DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CUR-
    RENCY>
    <DS_MERCHANT_PAN>4548812049400004</DS_MER-
    CHANT_PAN>
    <DS_MERCHANT_EXPIRYDATE>2012</DS_MERCHANT_EX-
    PIRYDATE>
    <DS_MERCHANT_CVV2>123533</DS_MERCHANT_CVV2>
```

```
    <DS_MERCHANT_TRANSACTIONTYPE>A</DS_MERCHANT_
    TRANSACTIONTYPE>
    <DS_MERCHANT_TERMINAL>2</DS_MERCHANT_TERMINAL>
    <DS_MERCHANT_IDENTIFIER>DE021281B7303F-
    3C3B2083A2BB150C21E6574946</DS_MERCHANT_IDEN-
    TIFIER>
    </DATOSENTRADA>
    <DS_SIGNATUREVERSION>HMAC_SHA256_V1</DS_SIGNA-
    TUREVERSION>
    <DS_SIGNATURE>GKPI63NIDOE7NKK+NVVLUVQYWWDHYQB-
    WMVCQKOHTV6I=</DS_SIGNATURE>
</REQUEST>
```

2 - 2018