TPV-Virtual

Apple Pay, Merchant Installation Guide

Issue: 2.0
Date: 08/03/2021
Reference: RS.SO.SAD.MAN.0022

Redsys, Servicios de Procesamiento, S.L. – Calle Francisco Sancha, 12 – 28034 Madrid (Spain)
www.redsys.es

Authorisation and version control

Version	Date	Concerns	Brief description of the change
1.0	21/04/22	ALL	Initial version of the document
1.1	17/02/20	Enrolment	Additional information about Merchant ID and Payment Processing Certificate
2.0	08/03/2021	ALL	Update procedure
2.1	30/12/2021	Point 3	A new way for the merchant to connect to the Virtual POS by sending decoded data is added.

TABLE OF CONTENTS

1. Introduction

The purpose of this document is to define and facilitate the integration of merchants with Apple Pay in the Redeyes Virtual POS, providing the information to register the merchant in the Apple Pay platform and indicating the integration process that the merchant has to follow to offer this payment method to its clients.

** This integration can only be conducted in the production environment of the Virtual POS.

2. Integration with Apple Pay InApp

The merchant must register in the Apple Pay platform, as a result, a merchant identifier (merchantId) and a certificate whose private key will be used for the subsequent decryption of the payment method data will be generated.

2.1 Merchant Enrolment with Apple Pay

- **Creation of merchantID**. Access the Apple Console and create a new MerchantId that identifies the Merchant. To register a Merchant ID:

1. Select Certificates, Identifiers & Profiles.
2. Under identifiers, select Merchant IDs.
3. Click on the add button (+) in the top right corner.
4. Enter a description and an identifier and click continue.
5. Review the settings and click Register.
6. Click on Done.

More information at
https://developer.apple.com/documentation/applepaywebmerchantregistrationapi/registering_with_apple_pay_and_applying_to_use_the_api

- **Creation of Merchant Identity Certificate**.

1. An RSA2048 private key and a certificate signing request (CSR) with said key are generated using the OpenSSL tool:

   openssl req -new -newkey rsa:2048 -nodes -keyout merchant_id.key -out merchant_id.csr -subj '/O=NombreComercio /C=ES'
2. This action will generate a merchant_ir.csr file that must be used to generate the final certificate by Apple
3. In the member centre, click on “Certificates, Identifiers & Profiles.” Click on “Identifiers” and then select “Merchant IDs” from the drop-down menu on the right. Click on the Merchant ID in the table.
4. At the bottom of the page, click on “Create Certificate” under the heading “Apple Pay Merchant Identity Certificate”
5. Upload the *merchant_id.csr* file generated in point 3.2 and click “Continue”

6. Click on “Download” to download a file called *merchant_id.cer*
7. Convert the *.cer file to a *.pem file using the following command in the OpenSSL tool:

   openssl x509 -inform der -in merchant_id.cer -out merchant_id.pem

- **Generation of the Certificate Signing Request (CSR)**. Once the Merchant Id and the Merchant Identity Certificate have been generated, it is necessary to generate a Certificate Signing Request used for decryption of the messaging from a private key. This Private key will be the one that the merchant must configure in the Virtual POS Management Portal (see point 2.2.1).

1. Generate key pair to elliptic curve key file with OpenSSL tool:

   openssl ecparam -genkey -name prime256v1 -out ecckey.key

2. Format the key to PKCS8 format:

   openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in ecckey.key -out ecckey_pkcs8.key

   ***The contents of the resulting file ecckey_pkcs8.key must be entered, together with the merchantId, in the Virtual POS administration portal.**

3. Generate CSR from the key pair in the key file

   openssl req -new -sha256 -key ecckey.key -out ecccertreq.csr -subj /CN=www.dominiocomercio.com

4. In the member centre, select Certificates, Identifiers & Profiles.
5. Under identifiers, select Merchant IDs.
6. Select a Merchant ID from the list and click Edit.
7. In the Payment Processing Certificates section, click Create Certificate. Follow the instructions to obtain or generate your Certificate Signature Request (CSR) and click Continue.
8. Click Choose File, select your CSR (*ecccertreq.csr*) and click Generate.
9. Download the certificate by clicking Download and click Done.

More information at https://help.apple.com/developer-account/#/devb2e62b839?sub=devf31990e3f

2.2 Integration with the Virtual POS

In the authorisation transaction sent to the Virtual POS, the following additional parameters must be added:

DS_XPAYDATA	*10000 / A-N*	**Required. Field that includes the value of the "paymentData" object sent by Apple in Hexadecimal format**
DS_XPAYTYPE	*10 / A-N*	Required. Fixed value "Apple"
DS_XPAYORIGEN	*5 / A-N*	Required. Origin of the request. The possible values are: - WEB - InApp

In the Payment Methods section, the "ApplePay" option must be selected: Bold indicates the object to be reported in Hexadecimal format.


```
{
    "token":{
        "paymentData":{
            "data":"CZOOPbi/R7UBdW3as7T...0YYWuQ1iZhvtjAfx+A==",
            "signature":"MIAGCSqGSIb3DQEHAqCA...3gAtcDwfdZIAAAAAAAA=",
            "header":{
                "publicKeyHash":"i4BsP3h7AdaM3DU30UA2pucLcPYT1J9bAj3gi8eAOzw=",
                "ephemeralPublicKey":"MFkwEwYHKoZIzj0CAQYI...OyHpyVnRod+CpBTMxQ==",
                "transactionId":"a9bf8e71ca58173d42af5f24b57ce047528fb285759b296c12f3d8f2e5926644"
            },
        "version":"EC_v1"
        },
        "paymentMethod":{
        "displayName":"Visa 0492",
        "network":"Visa",
        "type":"debit"
    },
    "transactionIdentifier":"A9BF8E71CA58173D42AF5F24B57CE047528FB285759B296C12F3D8F2E5926644"
    }
}
```

Below is an example of a request to the Virtual POS

<DATOSENTRADA>

<DS_MERCHANT_AMOUNT>42</DS_MERCHANT_AMOUNT>

<DS_MERCHANT_ORDER>813734</DS_MERCHANT_ORDER>

<DS_MERCHANT_MERCHANTCODE>999008881</DS_MERCHANT_MERCHANTCODE>

<DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CURRENCY>

<DS_MERCHANT_TRANSACTIONTYPE>0</DS_MERCHANT_TRANSACTIONTYPE>

<DS_MERCHANT_TERMINAL>871</DS_MERCHANT_TERMINAL>

**<DS_XPAYDATA>7B2276657273696F6E223A2245435F7631222C2264617461223A224875
3175315039414370305 94A53316C2B57746B6D6330626857544838557855484259617446
6D365558363451597859666 24D4846757964674D4136414B465556676C693869746C462
B7965495A704532572B686E6E796A56446F74714644472F483850425579526A737847436
F782B4B71726741756A504B3146565578595448304 72F7A427862763556506174316C344
54E7132635A6D6A686251377341504C7A39584E415039756C3463384668697834652F473
848447A6A5139307770616A6D4E57342B4A71444B72516F7A504B4B6B56723661617272
76512B4746594230594976447458744251586A57494E424B774B524B4E524A6B532F6445
507A736770456245684374333336426E4D53493555306C7131466B3873724676412B554B
526B77466572446E617479465A4F764A6B5942624D424C30434531726C6C6842466E667
0595A6F70772B764652725A356749413531484E2F52486E6D4F4C313839767A6B30652F5
0514F37412B2B54754F.....</DS_XPAYDATA>**

<DS_XPAYTYPE>Apple</DS_XPAYTYPE>

<DS_XPAYORIGEN>InApp</DS_XPAYORIGEN>

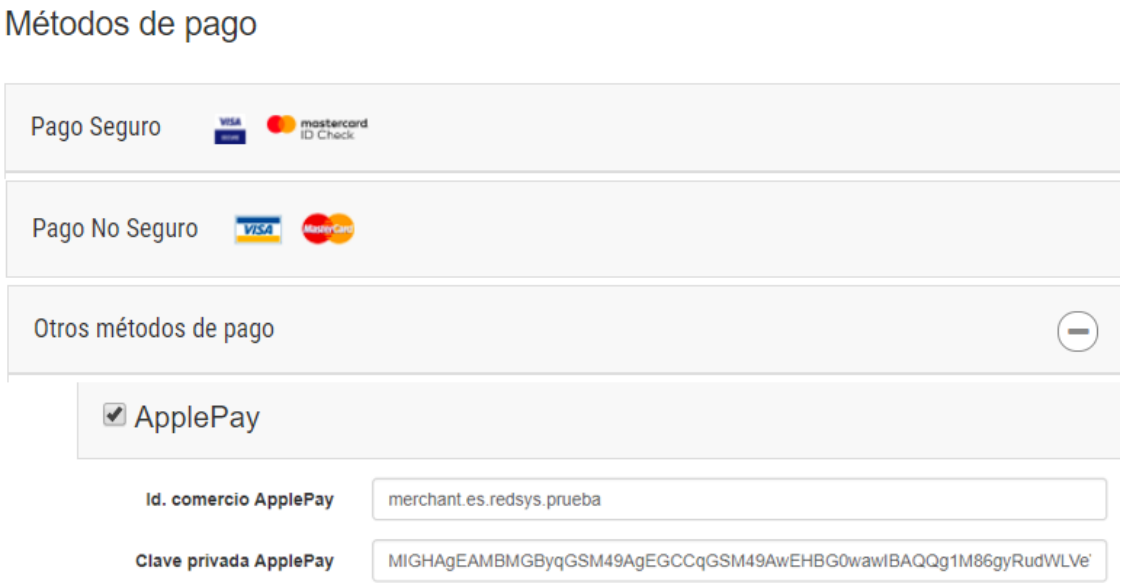
</DATOSENTRADA>

2.2.1 Configuration in the administration portal

The entity must configure the merchant credentials in the SIS Virtual POS Administration Portal, in the terminal configuration section.

In the *Payment Methodssection*, select the "ApplePay" option:

It will be selected in the "Other payment methods" submenu. Once the payment method has been selected, the fields corresponding to the merchant ID and the private key generated by the merchant (PKCS8 format) will be filled in in the enrolment process:



3. Advanced feature: Sending decrypted data by the Merchant

In this case, the merchant will be responsible for decrypting the data, obtaining the necessary information.

https://developer.apple.com/library/archive/documentation/PassKit/Reference/PaymentTokenJSON/PaymentTokenJSON.html

Key	Value	Description
applicationPrimaryAccountNumber	string	Account number specific to the card device that finances this transaction.
applicationExpirationDate	date as a string	Card expiry date in YYYYMMDD format.
onlinePaymentCryptogram	A Base64 encoded string	Online payment cryptogram, as defined by 3-D Secure.
eciIndicator	string	ECI indicator, as defined by 3-D Secure.

This data will be sent to the Virtual POS in the parameter **Ds_**XPayDecodedData, in Json format and with the following relationship:

```
{
    "cryptogram":"XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
"eciInd":"XX",
    "expirationDate":"AAMM",
    "token": "XXXXXXXXXXXXXXXXXX"
}
```

cryptogram = Reference to onlinePaymentCryptogram.

eciInd = Reference to the eciIndicator.

expirationDate = Reference of the field applicationExpirationDate. It has to be sent in **YYMM** format.

token = Reference to the applicationPrimaryAccountNumber field.

In the authorisation transaction sent to the Virtual POS, the following additional parameters must be added:

Parameter	Value
Ds_XPayDecodedData	JSON object with required information
Ds_XPayType	Apple
Ds_XPayOrigen	InApp - If the integration is on a mobile App WEB - If the integration is Web

Below is an example of a request to the Virtual POS:

```
<DATOSENTRADA>
<DS_MERCHANT_AMOUNT>42</DS_MERCHANT_AMOUNT>
<DS_MERCHANT_ORDER>813734</DS_MERCHANT_ORDER>
<DS_MERCHANT_MERCHANTCODE>999008881</DS_MERCHANT_MERCHANTCODE>
<DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CURRENCY>
<DS_MERCHANT_TRANSACTIONTYPE>0</DS_MERCHANT_TRANSACTIONTYPE>
<DS_MERCHANT_TERMINAL>871</DS_MERCHANT_TERMINAL>
<DS_XPAYDECODEDDATA>{"cryptogram":
"AgAAAAABk4DWZ4C28yUQAAAAAA=","eciInd":
"05","expirationDate":"2612","token":
"489537******3478"}</DS_XPAYDECODEDDATA>
<DS_XPAYTYPE>Apple</DS_XPAYTYPE>
<DS_XPAYORIGEN>InApp</DS_XPAYORIGEN>
</DATOSENTRADA>
```