iPayment Integration - One-time Credit Card

One-time Credit Card allows for easy integration with iPayment.
When using the one-time Credit Card integration the credit card token will be associated with a specific document instead of the business partner.

The integration is done using a user defined table (UDT) where the third party system writes data.
iPayment will check the user defined tables periodically and fill the iPayment tables using the data.

Contents

Integration table description

The user defined table is created by iPayment when run in the database and have the following structure:

Code (SAP default field)	Unique code
Name (SAP default field)	Unique name
DocEntry	DocEntry of the document in SAP this transaction is linked with
ObjectType	ObjectType of the document in SAP this transaction is linked with
Identifier	The identifier returned by the gateway to represent this customer (Customer Profile Id). This is required, as it will be used when settling and refunding a transaction. **Secure Trading:** Secure trading does not return this. Instead, please create a unique identifier (GUID) and send it to the gateway in the field "orderreference" when adding the card to the Card Store. **CyberSource:** CyberSource does not return this. Leave blank for CyberSource. **Authorize.NET/Secure Trading:** iPayment verifies that the card exist on the gateway using the identifier before the card is saved into the iPayment tables.
Credit Card Identifier	The token created by the gateway to represent the Credit Card. This is required as it will be used when doing a re-auth or settling without an existing auth. **CyberSource:** This is the data stored in the tag "subscriptionID"
Credit Card Type	Valid values for this field is found below pr. gateway: **Authorize.NET:** **Valid values:** Visa, MasterCard, American Express, Discover, JCB, Diners Club **Secure Trading:** **Valid values:** VISA, DELTA, PURCHASING, ELECTRON, MASTERCARD, MASTERCARDDEBIT, MAESTRO, DINERS, DISCOVER, JCB, AMEX, VPAY **CyberSource**: **Valid values:** 001, 002, 003, 004, 005, 007 (Visa, MasterCard, AmericanExpress, Discover, DinersClub, JCB)
Expiration date	String in format: MM/YYYY
Authorization transaction reference	The reference returned by the gateway when creating the authorization.

	This is required as it will be used when settling the auth. Leave empty if no authorization was made.
Authorized amount	The amount that was authorized on the Credit Card. This is required as it allows for partial authorizations. Set to 0 if no authorization was made.
Gateway Type	Enum value for the gateway type used. 1 = Secure Trading, 2 = Authorize.NET, 3 = CyberSource
Version	Always Value: 1
Timestamp	String in format: YYYY-MM-dd HH:mm:ss
ImportStatus	0 for not imported. 1 for imported. 2 for error.
ImportMessage	Will contain data if the import gave an error

Integration table definition:

Tablename: [@BOY_E0_OTCINTE]

Code	nvarchar(30)
Name	nvarchar(30)
U_DOCENTRY	int
U_OBJECTTYPE	nvarchar(254)
U_IDENTIFIER	nvarchar(254)
U_CREDITIDENTIFIER	nvarchar(254)
U_CARDTYPE	nvarchar(20)
U_EXPIRATION	nvarchar(7)
U_AUTHREF	nvarchar(254)
U_AUTHAMOUNT	numeric(19, 6)
U_GATEWAY	int
U_VERSION	int
U_TIMESTAMP	nvarchar(254)
U_IMPORTSTATUS	int
U_IMPORTMSG	nvarchar(254)

CyberSource additional information

iPayment uses the “recurringSubscriptionInfo” object to tokenize the Credit Card data. The token should be generated with “frequency” = “on-demand” and “paySubscriptionCreateService” = “true”. The returned “subscriptionID” should be saved into the integration table.

Adding the Credit Card and getting the token is possible using the XML API
(The C# example below is missing the header + soap wrapper).

```
XElement("billTo",
    XElement("firstName", creditCardRequest.FirstName),
    XElement("lastName",creditCardRequest.LastName),
    XElement("street1", creditCardRequest.Street),
    XElement("city",creditCardRequest.City),
    XElement("state",creditCardRequest.State),
    XElement("postalCode",creditCardRequest.ZipCode),
    XElement("country",creditCardRequest.Country),
    XElement("email",creditCardRequest.Email)
),
XElement("purchaseTotals",
    XElement("currency",creditCardRequest.Currency)
),
XElement("card",
    XElement("accountNumber",creditCardRequest.CardNumber),
    XElement("expirationMonth",creditCardRequest.ExpiryDateMonth),
    XElement("expirationYear",creditCardRequest.ExpiryDateYear),
    XElement("cvNumber", creditCardRequest.SecurityCode),
    XElement("cardType", GetCardTypeFromEnum(creditCardRequest.CardType))
),
XElement("recurringSubscriptionInfo",
    XElement("frequency","on-demand")
),
XElement("paySubscriptionCreateService", new XAttribute("run","true"))
```

Authorizing on the returned “subscriptionID” is possible using the XML API.
The returned ”requestID” should be added to the “Authorization transaction reference” field in the integration table.

(The C# example below is missing the header + soap wrapper).

```
XElement("purchaseTotals",
    XElement("currency",request.Currency),
    XElement("grandTotalAmount", GetDecimalAsString(request.Amount))
),
XElement("recurringSubscriptionInfo",
    XElement("subscriptionID", request.CreditCardTransactionReference)
),
XElement("ccAuthService", new XAttribute("run","true")),                ),
                XElement("ccAuthService", new XAttribute("run","true")),
```

Example Authorize.NET

A record created by a third-party system using Authorize.NET would look like (Where the code and name is unique as required by SAP):

Code (SAP default field)	1
Name (SAP default field)	1
DocEntry	132
ObjectType	17
Identifier	38220908
Credit Card Identifier	34738935
Credit Card Type	VISA
Expiration date	11/2018
Authorization transaction reference	34134324
Authorized amount	732.50
Gateway Type	2
Version	1
Timestamp	2015-11-19 08:55:22
ImportStatus	0
ImportMessage	

Example CyberSource:

Code (SAP default field)	1
Name (SAP default field)	1
DocEntry	132
ObjectType	17
Identifier	
Credit Card Identifier	347389353242342
Credit Card Type	001
Expiration date	11/2020
Authorization transaction reference	341343242342342
Authorized amount	832.50
Gateway Type	3
Version	1
Timestamp	2015-11-19 08:55:22
ImportStatus	0
ImportMessage	