

int_altapay

Version 17.1.0



Table of Contents

1	SUMMARY	4
2	RELEASE HISTORY	4
3	COMPONENT OVERVIEW	4
3.1	FUNCTIONAL OVERVIEW	4
3.2	USE CASE	4
3.3	ACTORS	4
3.4	LIMITATIONS, CONSTRAINTS	5
3.5	COMPATIBILITY	6
3.6	PRIVACY, PAYMENT	6
4	IMPLEMENTATION GUIDE	6
4.1	SETUP	6
4.2	METADATA IMPORT & CUSTOM PREFERENCES	6
4.3	CONFIGURATION	8
4.4	CUSTOM CODE	11
4.5	EXTERNAL INTERFACES	15
5	CREDIT CARD TOKENIZATION	15
5.1	CONTACT ALTA PAY TO ENABLE THE CREDIT CARD TOKEN IN YOUR TERMINAL	15
5.2	SETUP BUSINESS MANAGER	15
6	RECONCILIATION SETUP	17
6.1	HOW TO SET UP THE RECONCILIATION IDENTIFIER	17
7	TESTING	20
7.1	SUCCESSFUL CARD PAYMENT	20
7.2	FAILED CARD PAYMENT	21
7.3	3D SECURE CARD PAYMENT	21
7.4	3D SECURE FAILED CARD PAYMENT	21
7.5	SUCCESSFUL ALTERNATIVE PAYMENT	22
7.6	FAILED ALTERNATIVE PAYMENT (CUSTOMER CANCEL)	22
7.7	FAILED ALTERNATIVE PAYMENT	22
7.8	SUCCESSFUL ALTERNATIVE PAYMENT NOTIFICATION	23
7.9	FAILED ALTERNATIVE PAYMENT NOTIFICATION	23
7.10	FRAUD CHECKING (ACCEPTED CREDIT CARD)	23
7.11	FRAUD CHECKING (DENIED CREDIT CARD)	23
8	OPERATIONS, MAINTENANCE	24
8.1	DATA STORAGE	24
8.2	AVAILABILITY	24
8.3	SUPPORT	24

9	<u>USER GUIDE</u>	24
9.1	ROLES, RESPONSIBILITIES	24
9.2	BUSINESS MANAGER	24
9.3	STOREFRONT FUNCTIONALITY	24
10	<u>KNOWN ISSUES AND LIMITATIONS</u>	25
10.1	PLANNED CHANGES	25

1 Summary

This cartridge enables a Demandware shop to use AltaPay as the Payment Service Provider (PSP). To enable AltaPay as the PSP, the merchant needs to:

1. Sign a contract with AltaPay
2. Install the `int_altapay`, `int_altapay_controllers` and `int_altapay_pipelines` cartridges
3. Import 'Metadata_AltaPay.xml'
4. Conduct tests of the payment flow before going live

Before going live, the merchant needs to conduct tests of the payment flow. AltaPay provides a test setup for accepting card and alternative payments. The merchant can also use AltaPay's backend for captures, refunds and releases of payments.

2 Release History

Version	Date	Changes
15.1.0	2015.05.22	Initial release
15.1.0	2015.11.10	Improved user guide (images added etc.)
15.1.0	2016.12.9	Updated documentations
17.1.0	2017.01.16	Instructions about the updated plugin version (that uses controllers instead of pipelines)
17.1.0	2017.04.07	Cartridges information updated.

3 Component Overview

3.1 Functional Overview

The idea of the payment gateway is to allow your customers to perform secure payments without the feeling that they are leaving your web shop. This is possible because AltaPay proxy the payment page from your website – keeping layout and visual identity. The AltaPay Payment Gateway will inject a payment form into the payment page, which reflects the payment method (Credit Card, Bank Payment, etc.).

3.2 Use Case

The below described use case covers the scenarios a merchant, customer and AltaPay experience when making a payment.

3.3 Actors

- **Customer:** The buyer and payer of items at the web shop
- **Merchant:** Provides the web shop and items to be sold

- **AltaPay:** Processes payment information and verifies payment information provided by the customer with help from an acquirer

3.3.1 General payment flow

1. The customer visit the merchant's web shop and add items to the basket
2. Customer is ready to pay and select preferred payment method
3. Customer is redirected to AltaPay where he enters required payment details (Card number with/without 3D secure, PayPal account etc.)
4. Payment details are sent to AltaPay where they are processed with the relevant acquirer
5. After processing the customer returns to the web shop, where they are informed about the outcome

3.3.2 Card payment flow

Preconditions: Customer has added items in the basket and entered billing and shipping details

1. Customer select card as payment method
2. Customer is presented with the payment page where card details can be typed
3. If the card is enrolled for 3D Secure - the card issuer's bank present a page where the customer provides their 3D Secure information
 - a) if the 3D Secure details are correct and the issuer approves the transaction, the payment is complete. The customer is informed and the merchant can process the order.
 - b) in case the 3D Secure details are incorrect or the issuer declines the transaction for other reasons, the customer is redirected to the basket and requested to select payment method again.

3.3.3 Alternative payment flow

Preconditions: Customer has added items in the basket and entered billing and shipping details

1. Customer select an alternative payment method
2. The customer is redirected to the alternative payment provider and will be requested to login. After the customer has logged in, he must accept the pending payment.
3. When the payment has been accepted, the customer is redirected back to the merchant's web shop where the customer is informed that the payment is complete and the merchant is able to process the order.
 - In case the payment could not be verified, but have not been rejected, the payment end in a status called open (e.g. due to the provider performing a manual fraud review).
Here the customer is also shown a confirmation page but with a notice that the payment has not been confirmed. The merchant will also be able to see this, as the order has the confirmation status: 'Not confirmed'.
If the payment is declined after verification, the status of the order will change to 'Cancelled'. If the payment is confirmed, the status of the order will change to 'Confirmed'.

3.4 Limitations, Constraints

- The merchant will need a AltaPay test/production terminal
- The merchant must implement their own look and feel on the payment page - otherwise a standard page is shown
- Alternative payments which cannot be preauthorized is shown as cancelled, instead of failed
- For asynchronous payments, which are completed as a successful or declined status at a later stage, the merchant is responsible to notify the customer

3.5 Compatibility

Based on Demandware and SiteGenesis: 17.1 (compatibility mode: 16.2)

Tested on Demandware and SiteGenesis: 103.1.5

3.6 Privacy, Payment

Information about payment method, debit/credit card data, items, shipping/billing addresses and amount is sent to AltaPay.

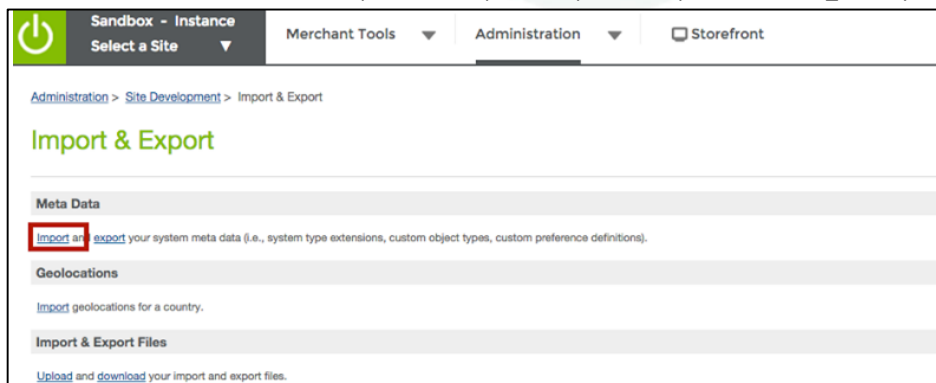
4 Implementation Guide

4.1 Setup

1. Download the 'int_altapay', 'int_altapay_controllers' and 'int_altapay_pipelines' cartridges from [Demandware LINK marketplace](#)
2. Install the cartridges using Demandware UX-studio. Remember to include the cartridges in the cartridge path
 - I. If your store is going to use controllers, add 'int_altapay' and 'int_altapay_controllers' to the path
 - II. If not, add 'int_altapay' and 'int_altapay_pipelines'

4.2 Metadata import & Custom preferences

1. Go to: Administration → Site Development → Import & Export → Import 'Metadata_AltaPay.xml'



2. Go to: Merchant Tools → Site Preferences → Custom Site Preferences → AltaPay. This is the place where the merchant can access the AltaPay custom preferences site.
3. The table below describe each field for the AltaPay custom preference site in more details. In Demandware the table also contain a third column stated with default values where applicable.

Preference	Description
AltaPay base production URL	URL for the production environment
AltaPay base test URL	URL for the test environment

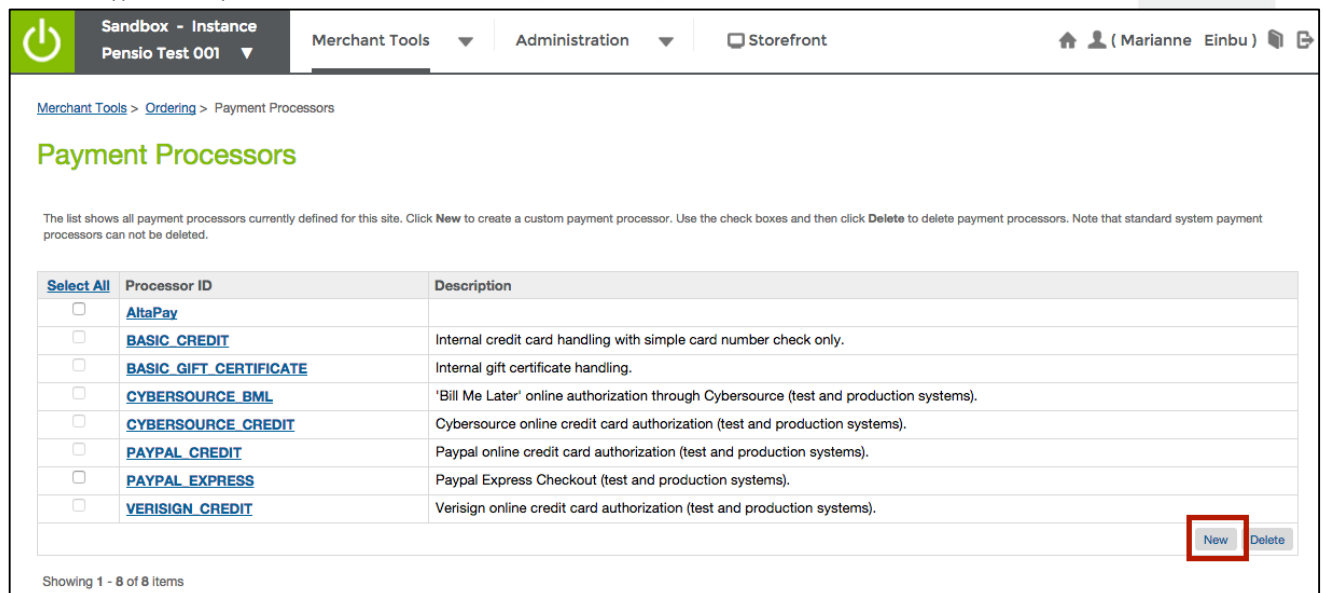
AltaPay Production Username	Username for the production environment
AltaPay Production Password	Password for the user in production environment
AltaPay Test Username	Username for the test environment
AltaPay Test Password	Password for the user in test environment
AltaPay Test Mode	This toggles whether production or test settings are used for payment processing
AltaPay terminals	<p>Mapping payment methods in Demandware and AltaPay's terminals. The attribute 'id' must correspond with the payment method added in: Merchant Tools → Ordering → Payment Methods. The attribute 'name' is the name of the AltaPay terminal. A terminal can only contain one payment method and one currency, but it is possible to add the relevant terminals. The setting has to be structured as shown below.</p> <pre> { "terminals" : { "production" : { "en_GB" : [{ "id": "ALTAPAY_CC_EUR", "name" : "terminal name" }] }, "test" : { "en_GB" : [{ "id": "ALTAPAY_CC_EUR", "name" : "terminal name" }] } } } </pre>
AltaPay payment page URL	<p>URL for controlling the payment form page which is shown to the customer.</p> <p>It is possible to customize the payment page. Go to Demandware UX-studio → Templates → Default → altapay and change the callbackform.isml. Javascript and other dynamic content from the payment page will be removed for security reasons.</p>
AltaPay payment success URL	When a payment is accepted, this URL validates the data received from AltaPay.
AltaPay payment fail URL	In case a payment fails. This can be due to incorrect card details, declined by the bank etc.

AltaPay payment open URL	To support an asynchronous payment (e.g. wallet payments) where the provider not always accept the payment upfront. To indicate this event an open payment contains the confirmation status 'Not confirmed'.
AltaPay payment notification URL	In case a payment has not returned an answer (e.g. customer closes window prior to returning to the shop), or when an open payment is accepted/declined. When an answer arrives, this URL is called. This does not apply to card payments.
AltaPay redirect page URL	This URL is used when the customer is redirected to a third party (e.g. 3D Secure) to inform the customer about the redirection. A default non-branded page is shown if nothing else is stated.
AltaPay timeout	This setting determines the timeout for communication with the AltaPay backend. Do not change the value.

4.3 Configuration

4.3.1 Add terminal

1. Setup 'AltaPay' as a payment processor. This can be done here: Merchant Tools → Ordering → Payment Processors → Click 'New' → Type 'AltaPay' next to ID



Payment Processors

The list shows all payment processors currently defined for this site. Click **New** to create a custom payment processor. Use the check boxes and then click **Delete** to delete payment processors. Note that standard system payment processors can not be deleted.

Select All	Processor ID	Description
<input type="checkbox"/>	AltaPay	
<input type="checkbox"/>	BASIC_CREDIT	Internal credit card handling with simple card number check only.
<input type="checkbox"/>	BASIC_GIFT_CERTIFICATE	Internal gift certificate handling.
<input type="checkbox"/>	CYBERSOURCE_BML	'Bill Me Later' online authorization through Cybersource (test and production systems).
<input type="checkbox"/>	CYBERSOURCE_CREDIT	Cybersource online credit card authorization (test and production systems).
<input type="checkbox"/>	PAYPAL_CREDIT	Paypal online credit card authorization (test and production systems).
<input type="checkbox"/>	PAYPAL_EXPRESS	Paypal Express Checkout (test and production systems).
<input type="checkbox"/>	VERISIGN_CREDIT	Verisign online credit card authorization (test and production systems).

Showing 1 - 8 of 8 items

To add a terminal, the merchant must do two things:

1. Add a new payment method and update the custom site preferences to correspond with the payment method and AltaPay terminal. Please note that the terminal(s) needs to be created by AltaPay first.

- Go to: Merchant Tools → Ordering → Payment Methods → Click 'New' → AltaPay as 'Payment Processor' → click 'Apply'
- Update the ID and Name for the payment method. The ID must begin with 'ALTAPAY_' and it is recommended to follow this standard: 'ALTAPAY_<payment_method>_<currency>' e.g. 'ALTAPAY_CC_EUR'. The name is what is shown on the web page and make the customer able to choose their preferred payment method.

Payment Methods

Payment methods are managed here. To create a new payment method, click the **New** button. To remove a payment method click the remove icon in the payment method row. The default payment methods cannot be removed, and their IDs cannot be changed. When you select the CREDIT_CARD payment method, credit/debit cards can be reordered through drag and drop.

Language: Default

ID	Name	Enabled	Sort Order
ALTAPAY_CREDITCARDS	Credit card	Yes	3
ALTAPAY_CREDITCARDS_DKK	Visa Dankort	No	8
ALTAPAY_CREDITCARDS_GBP	Credit card	No	7
ALTAPAY_PAYPAL	E-wallet	Yes	4
BANK_TRANSFER	Bank Transfer	No	5
BML	Bill Me Later	No	6
CREDIT_CARD	Credit Card	No	2
GIFT_CERTIFICATE	Gift Certificate	No	1
New Payment Method - 11/10/15 8:16:22 am		No	9

3. After step 2 a new row appear. Rename 'ID': E.g. 'ALTAPAY_CREDITCARDS_EUR'.
4. Rename 'Name': E.g. 'Credit Card'

ALTAPAY_CREDITCARDS Details

Description:

Image: [HTML Editor](#) [Select](#)

Payment Processor: 1. Select AltaPay as Payment Processor

Countries: [Edit](#)

Customer Groups: [Edit](#)

Min/Max Payment Ranges: [Min/Max Payment Ranges](#)

DKK to

2. Click 'APPLY' [Apply](#) [Cancel](#)

- The second step is to update the custom preference 'AltaPay terminals'.

Go to: Merchant Tools → Site Preferences → Custom Preferences → Click on 'AltaPay'.

Here you add the terminals which have been provided by AltaPay. The setting only works in JSON format. The JSON object 'terminals' consist of two variables: 'production' and 'test'. Each of these objects can have many variables (arrays) depending on the number terminals. Each of these arrays have the name after the character language code, that corresponds with the currency code that was added in the payment method. In the array, there is only one object per payment method with two elements - Demandware payment id and the AltaPay terminal name.

```
{
  "terminals" : {
    "production" : {
      "en_GB" : [
        { "id": "ALTAPAY_CC_GBP", "name" : "Shopname CC gbp" }
      ],
      "dk_DA" : [
        { "id": "ALTAPAY_CC_DKK", "name" : "Shopname CC dkk" }
      ]
    },
    "test" : {
      "en_GB" : [
        { "id": "ALTAPAY_CC_GBP_TEST", "name" : "Test Terminal gbp " }
      ],
      "dk_DA" : [
        { "id": "ALTAPAY_CC_DKK_TEST", "name" : "Test Terminal dkk" }
      ]
    }
  }
}
```

4.3.2 Select language

As part of the setup, the language selection for the check-out process is also on the check list.
Go to: Merchant Tools → Site Preferences → Locales → select the web shops local language.

AltaPay supports the following languages:

cs	Czech
da	Danish
de	German
en	English
es	Spanish
fi	Finnish
fr	French
ja	Japanese
lt	Lithuanian
nl	Dutch

no	Norwegian
nb*	Norwegian (Bokmål) - converted to no
nn*	Norwegian (Nynorsk) - converted to no
pl	Polish
sv	Swedish
th	Thai
tr	Turkish
zh	Chinese
ee*	Estonian - converted to et
et	Estonian
it	Italian
pt	Portuguese
ru	Russian

If the merchant uses an unsupported language the payment page is shown in English as default.

4.4 Custom Code

We have developed some code, which you need to implement to be able to inform the customer if a payment has failed or has not been completed. The merchant also needs to inform the customer when the payment information is complete.

If implementing the web shop from scratch, you will probably create three projects using the UX Studio plugin for the Eclipse IDE. The projects names will be:

- `<project>_core`
- `<project>_controllers`
- `<project>_pipelines`

Where `<project>` can be any chosen name.

To create these projects using Eclipse, go to:

New > Other > UX Studio > SiteGenesis Storefront cartridges

The 'Cartridges path' in Demandware must also be setup. Using Demandware Business Manager, follow the instructions below:

- I. Go to: Administration > Sites > Manage Sites
- II. Click on the correct site
- III. Click on the 'Settings' tab
- IV. If your store is going to use controllers, set the 'Cartridges' path to:

```
int_altapay:int_altapay_controllers:<proj>_controllers:<proj>_core:<other cartridges, if necessary>
```

- V. If your store is going to use pipelines, set the 'Cartridges' path to:

```
int_altapay:int_altapay_pipelines:<proj>_pipelines:<proj>_core:<other cartridges, if necessary>
```

The order of the cartridges is important.

Seven files will have to be modified, as explained in the next sections. The locations of the files are:

File	Location
billing.isml	<proj_core>/cartridge/templates/default/checkout/billing
confirmation.isml	<proj_core>/cartridge/templates/default/checkout/confirmation
COSummary.js	<proj_controllers>/cartridge/controllers
COPlaceOrder.js	<proj_controllers>/cartridge/controllers
CreatePaymentInstrument.ds	int_altapay/cartridge/scripts/pipelet
UpdateErrorMessage.ds	int_altapay/cartridge/scripts/pipelet
UpdateOpenNotificationMessage.ds	int_altapay/cartridge/scripts/pipelet
AltaPay.js	int_altapay_controllers/cartridge/controllers

4.4.1 Error message

Add the code below to show an error message to the customer if a payment fails. Place the code by following this path:

1. Go to <proj_core>/cartridge/templates/default/checkout/billing/billing.isml or the merchant implementation of the billing page. The error message is an AltaPay error message and the merchant should implement its own error message, which can be done in the UpdateErrorMessage.ds script. An example can be found in the folder 'Custom code'.

```
<div class="altapay-errorMessage">
  <isif condition="{request.httpParameterMap.altapayErrorMessage != null}">
    <label>
      ${request.httpParameterMap.altapayErrorMessage}
    </label>
  </isif>
</div>
```

4.4.2 [Open message](#)

To inform that a payment is opened but not confirmed the code below needs to be added.

```
<isif condition="{request.httpParameterMap.altapayOpenMessage != null}">
  ${request.httpParameterMap.altapayOpenMessage}
</isif>
```

Place the code by following this path:

Go to <proj_core>/cartridge/templates/default/checkout/confirmation/confirmation.isml. We recommend to insert the code between:

```
<div class="confirmation-message">
...
</div>
```

and:

```
<div class="actions">
...
</div>
```

The merchant must implement its own logic for localizing messages for the customer, which is done in the `UpdateOpenNotificationMessage.ds`. An example can be found in the folder 'Custom code'.

4.4.3 [Update CreatePaymentInstrument.ds](#)

In the `CreatePaymentInstrument.ds` script there is a reference to "TestSite:checkout/Utils.ds" on line 23, which you need to change to fit the merchant's setup.

4.4.4 [Update COSummary.js](#)

The method `submit()` inside `COSummary.js` must be replaced.

1. Go to <proj_controllers>/cartridge/controllers/COSummary.js
2. Replace the method `submit()` by the version below

```
function submit() {
  // Calls the COPlaceOrder controller that does the place order action and any payment authorization.
  // COPlaceOrder returns a JSON object with an order_created key and a boolean value if the order was created successfully.
  // If the order creation failed, it returns a JSON object with an error key and a boolean value.
  var placeOrderResult = app.getController('COPlaceOrder').Start();
```

```

if (placeOrderResult.error) {
  start({
    PlaceOrderError: placeOrderResult.PlaceOrderError
  });
} else if (placeOrderResult.order_created) {

  if (request.custom.altapay_location != null) { // new code
    response.redirect(request.custom.altapay_location); // new code
  }
  else {
    showConfirmation(placeOrderResult.Order);
  }
}
}

```

4.4.5 Update COPlaceOrder.js

The method start() inside COPlaceOrder.js must be updated.

1. Go to <proj_controllers>/cartridge/controllers/COPlaceOrder.js
2. Substitute the last lines of the method start(). Follow the instructions below:

Substitute these last lines:

```

var orderPlacementStatus = Order.submit(order);
if (!orderPlacementStatus.error) {
  clearForms();
}
return orderPlacementStatus;

```

They must be replaced by this:

```

if (request.custom.altapay_location != null) {

  // Return without changing the order status to NEW. The order status will remain CREATED,
  // which means it can be recovered (the user can try to buy it again) if the gateway unauthorizes the order.

  return {
    Order: order,
    order_created: true
  };
}
else { // the order status will be changed to NEW
  var orderPlacementStatus = Order.submit(order);
  if (!orderPlacementStatus.error) {
    clearForms();
  }
  return orderPlacementStatus;
}
}

```

4.4.6 Update AltaPay.js

It is necessary to update the file int_altapay_controllers/cartridge/controllers/AltaPay.js. In lines 9 and 13, update the name of the cartridge (altapay_controllers) to match your controllers' cartridge name (<project>_controllers).

4.5 External Interfaces

The AltaPay cartridge communicates with AltaPay's backend where customer data etc. is sent, in order to verify a transaction. Banks and acquirers make the verification. AltaPay relays the response to the cartridge.

5 Credit card tokenization

It's possible to save the customer credit card after a successful transaction. The credit card number is saved securely inside AltaPay payments gateway. To enable this functionality, follow the steps below.

5.1 Contact AltaPay to enable the credit card token in your terminal

The credit card terminal must be setup to support credit card tokens. Also, the credit card form template must be set to `form_dynamic_div_with_save_cc`. This setup is done inside AltaPay payments gateway. Please contact AltaPay to setup your terminal.

If the terminal is setup correctly the customer will have the option to save the credit card information during checkout. Also, a previous saved credit card will appear with a mask in the checkout page, as shown in the image below.

FREE 2-Day SHIPPING FOR ORDERS OVER £300

salesforce commerce cloud

NEW ARRIVALS WOMENS MENS TOP SELLERS

Card Number
4170 00** **** 0006

Expiry Date
02
2017

CVC / CVV2

Typically located on the back of the card.

☒ Save my card details

FOLLOW US **ACCOUNT** **CUSTOMER SERVICE** **ABOUT**

5.2 Setup Business Manager

It is also necessary to setup your Business Manager instance:

- Go to: Administration > Site Development > System Object Types
- Click on the 'Profile' object
- Click on the tab 'Attribute Definitions'
- Click on 'New'
- Fill the 'ID' field with the value **ccToken**
- Fill the 'Display Name' field with 'Credit card token for AltaPay'
- 'Value Type' must be 'String'

See the image below for an example:

[Administration](#) > [Site Development](#) > [System Object Types](#) > [Customer Profile - Attribute Definitions](#) > Attribute Definition Details

Object Type 'Customer Profile' - Attribute Definition Details

On this page you can manage details of your attribute definition. Different options are available depending on the value type of your attribute. Click **Apply** to create a new attribute definition or save changes to existing attributes.

Select Language:
Default
Apply

ID*: ccToken

Display Name: Credit card token for AltaPay

Help Text:

Value Type: String
Note: Searchable via query framework.

Mandatory: ☐ *

Externally Managed: ☐

Value Unit:

Min Length: 0

Field Length:

Field Height:

Regex:

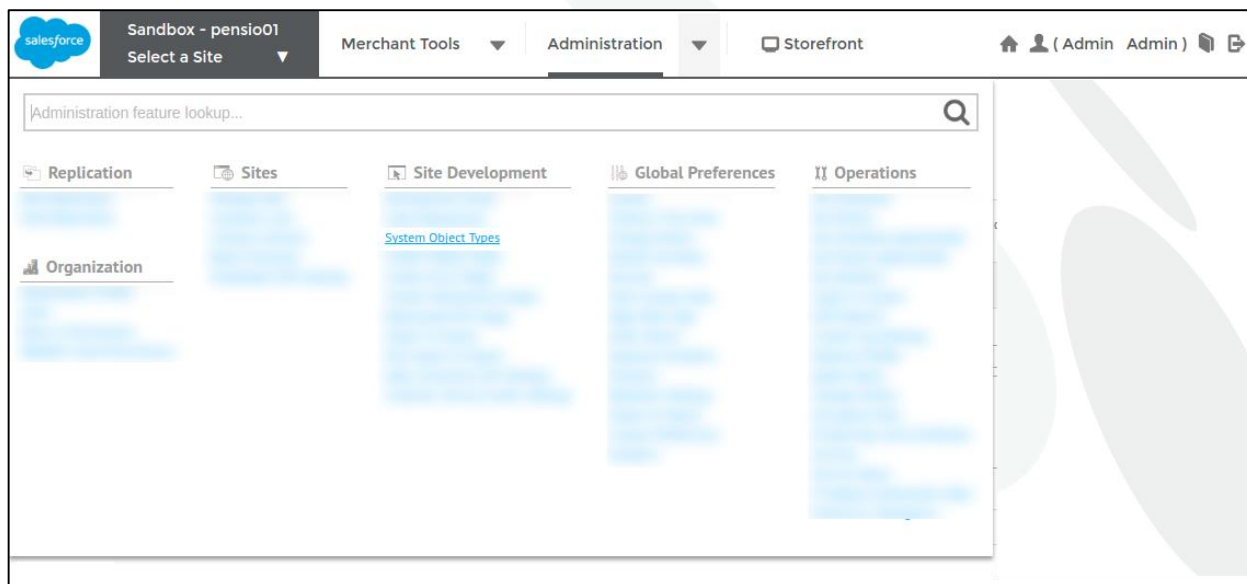
Apply
Reset

[<< Back](#)

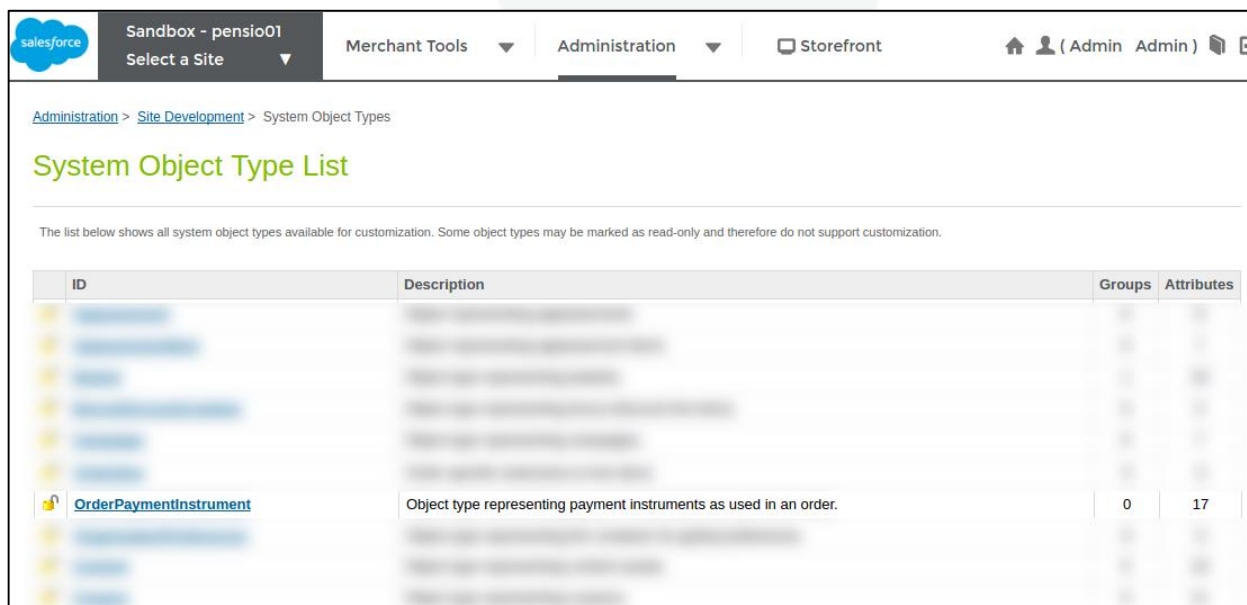
6 Reconciliation setup

6.1 How to set up the reconciliation identifier

1. Login into your Demandware account. Go to: Administration -> Site Development -> System Object Types.



2. Click on: OrderPaymentInstrument



3. Click on the tab Attribute Definitions. In the right corner, there is a button labeled "New". Click on it.

salesforce

Sandbox - pensio01

Select a Site

Merchant Tools

Administration

Storefront

(Admin Admin)

Administration > Site Development > System Object Types > Order Payment Instrument - Attribute Definitions

General

Attribute Definitions

Attribute Grouping

Object Type 'Order Payment Instrument'

This page lists the attribute definitions of your object type. Use the search to find attribute definitions by ID and name.

Click **New** to create new attribute definitions. Click **Delete** to delete existing attribute definitions.

Search Attribute Definitions

ID or Name:

Find

Select All	ID	Name	Type	Attribute Settings	Values	
<input type="checkbox"/>	UUID	UUID	String	*	0	Edit
<input type="checkbox"/>	bankAccountDriversLicense	Bank Account Drivers License	String		0	Edit
<input type="checkbox"/>	bankAccountDriversLicenseStateCode	Bank Account Drivers License State	String		0	Edit
<input type="checkbox"/>	bankAccountHolder	Bank Account Holder	String		0	Edit
<input type="checkbox"/>	bankRoutingNumber	Bank Routing Number	String		0	Edit
<input type="checkbox"/>	creationDate	Creation Date	Date+Time	*	0	Edit
<input type="checkbox"/>	creditCardExpirationMonth	Credit Card Expiration Month	Integer		0	Edit
<input type="checkbox"/>	creditCardExpirationYear	Credit Card Expiration Year	Integer		0	Edit
<input type="checkbox"/>	creditCardHolder	Credit Card Holder	String		0	Edit
<input type="checkbox"/>	creditCardIssueNumber	Credit Card Issue Number	String		0	Edit

New

Delete

Showing 1 - 10 of 17 items

Show All items

1

2

Next

>>

<< Back to List

- In the text field **ID**, enter **reconciliationIdentifier** and then click on 'Apply'.

salesforce

Sandbox - pensio01

Select a Site

Merchant Tools

Administration

Storefront

(Admin Admin)

Administration > Site Development > System Object Types > Order Payment Instrument - Attribute Definitions > Attribute Definition Details

Object Type 'Order Payment Instrument' - Attribute Definition Details

On this page you can manage details of your attribute definition. Different options are available depending on the value type of your attribute. Click **Apply** to create a new attribute definition.

Select Language: Default

Apply

ID:*

reconciliationIdentifier

Display Name:

Help Text:

Value Type:*

String

Apply

Cancel

<< Back

- Set the identifier on the payment request:

In the script file 'int_altapay\cartridge\scripts\pipelet\PaymentCreate.ds', in the function "execute", find this and remove the comments:

```
/*  
//add the sale_reconciliation_identifier  
parameterArr.push( ['sale_reconciliation_identifier', "Insert reconciliation identifier here" ].join( '=' ));  
*/
```

Replace "Insert the reconciliation identifier here" with the reconciliation identifier that is needed by the ERP system.

7 Testing

In general the merchant can use any card number when testing against the test gateway and they will be accepted. Designated card numbers to trigger different scenarios (3D Secure, failures etc.) can be found [here](#).

A Test bank is also available if the merchant needs to test PayPal, iDEAL, or other alternative payment methods.

If a merchant wants a test account, please read more [here](#).

Preconditions for the following test scenarios:

- Imported the Metadata_AltaPay.xml
 - Updated the 'Custom Site Preferences' with AltaPay user with API access rights
 - Added terminals for credit card and/or alternative payment
- Items in the Genesis web shops
- That the payment method for cards connects with a terminal that is configured to receive cards. An alternative payment method must be connected to a terminal that accepts that payment method.

7.1 Successful card payment

1. Add an item to the cart
2. Click 'View cart'
3. When shopping cart is shown, click on 'Checkout'
4. Select either Guest checkout or login
 - a) If guest checkout
 - I. Fill in the information
 - II. Click 'Continue'
 - III. Shipping information is shown. Fill in remaining information
 - IV. Select 'Credit card' as payment method and click 'Continue'
5. A Check summary is shown. Verify the details and click 'Submit order'
6. The payment page appears. Ensure it is a payment page for card payments.
7. Enter card details (use random numbers). Click 'complete'.
8. Verify that the summary page is shown with correct information and without any error message. Take a note of the order number.
9. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Confirmation Status' = confirmed.
10. Login in to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Check that the amount corresponds with the information in Business manager and ensure that the status of the payment is 'preauth'.

7.2 Failed card payment

1. Repeat step 1-6 in [Successful card payment](#)
2. Enter card details - use the following payment information:
 - a) Card number: 4180000000000566
 - b) Expire month: 05
 - c) Expire year: 2016
 - d) CVC: 444'
3. Click on 'complete'
4. Ensure that the basket is shown with the message 'Card Declined'. The merchant can change this error message and in most cases this is preferred to give the customer a user-friendlier decline message.
5. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Order Status' = failed. Take a note of the order number.
6. Login in to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Ensure that the status of the order is 'preauth_failed'.

7.3 3D Secure card payment

1. Repeat step 1-6 in [Successful card payment](#)
2. Enter card details - use the following payment information:
 - a) Card number: 4170000000000568
 - b) Expire month: 05
 - c) Expire year: 2016
 - d) CVC: 444
3. Click 'complete'. This will only work on the test gateway. In production, a real card with 3D secure enabled is required.
4. The user is redirected to the issuing bank 3D Secure confirmation page. Enter the correct validation information. If you are testing against the test gateway, a mock-up 3D Secure page is shown. Click 'Redirect'.
5. Verify that the summary page is shown with correct information and without any error message. Take a note of the order number.
6. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Confirmation Status' = confirmed.
7. Login in to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Check that the amount corresponds with the information in Business manager and ensure that the status of the payment is 'preauth' and that '3D Secure result:' is 'Successful'.

7.4 3D Secure failed card payment

1. Repeat step 1-6 in [Successful card payment](#)
2. Enter card details - use the following payment information:
 - a) Card number: 4170000000000568
 - b) Expire month: 05
 - c) Expire year: 2016
 - d) CVC: 444
3. The user is redirected to the issuing bank 3D Secure confirmation page. Enter the correct validation information. If you are testing against the test gateway, a mockup 3D Secure page is shown. Click 'Redirect'.
4. Ensure that the basket is shown with the message 'Card Declined'. This error message can be changed by the merchant and in most cases this is preferred to give the customer a more user friendly decline message.
5. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Order Status' = failed. Take a note of the order number.
6. Login in to <https://testgateway.pensio.com> and Locate the order by the order number via the search box in the top right corner. Ensure that the status of the order is 'preauth_failed'.

7.5 Successful alternative payment

1. Add an item to the cart
2. Click 'View cart'
3. When shopping cart is shown, click on 'Checkout'
4. Select either Guest checkout or login
 - a) If guest checkout
 - I. Fill in the information
 - II. Click 'Continue'
 - III. Shipping information is shown. Fill in remaining information
 - IV. Select preferred alternative payment method and click 'Continue'
5. A Check summary is shown. Check that the items correspond with what you have selected. Click on 'Submit order'.
6. Verify that the customer is redirected to the alternative payment provider webpage. Verify the pending payment.
7. If you are testing against the test gateway a mock-up for bank and alternative payment solutions will be shown. If that is the case, click 'Sign in' (No credentials needed) and 'Accept'.
8. Verify that the summary page is shown with correct information and without any error message. Take a note of the order number.
9. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Confirmation Status' = confirmed.
10. Login to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Check that the amount corresponds with the information in Business manager and ensure that the status of the payment is 'preauth' or 'bank_payment_finalized', depending on the acquirer.

7.6 Failed alternative payment (Customer cancel)

1. Repeat step 1-3 in [Successful alternative payment](#)
2. Verify that the customer is redirected to the alternative payment provider webpage. Verify the pending payment.
 - a) If you are testing against the test gateway a mockup for bank and alternative payment solutions will be shown. If that is the case, click 'Developer options' and 'Cancel'.
3. Ensure that the basket is shown with the message 'Cancelled by user'. The merchant can change the error message and in most cases this is preferred to give the customer a user-friendlier decline message.
4. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Order Status' = failed. Take a note of the order number.
5. Login to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Check that the amount corresponds with the information in Business manager and ensure that the status of the payment is 'epayment_cancelled' or 'preauth_failed' depending on the acquirer.

7.7 Failed alternative payment

1. Repeat step 1-3 in [Successful alternative payment](#)
2. Verify that the customer is redirected to the alternative payment provider webpage. Verify the pending payment.
 - a) If you are testing against the test gateway a mockup for bank and alternative payment solutions will be shown. If that is the case, click 'Developer options' and 'Declined'.
3. Ensure that the basket is shown with the message 'Declined'. The merchant can change this error message and in most cases this is preferred to give the customer a user-friendlier decline message.
4. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Order Status' = failed. Take a note of the order number.
5. Login to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Check that the amount corresponds with the information in Business manager and ensure that the status of the payment is 'epayment_cancelled' or 'preauth_failed' depending on the acquirer.

7.8 Successful alternative payment notification

1. Repeat step 1-3 in [Successful alternative payment](#)
2. Verify that the customer is redirected to the alternative payment provider webpage. Verify the pending payment.
 - a) If you are testing against the test gateway a mock-up for bank and alternative payment solutions will be shown. If that is the case, click 'Developer options' and 'Open (Opens in a new window)'. Do not close the test bank page.
3. Verify that a new window appear in the browser and contain a summary page with correct information. Also verify that a message indicating that the payment has not been confirmed is shown. Take a note of the order number.
4. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Order Status' = Not confirmed.
5. Go back to the test bank page and click 'Call success notification now'. It can take a couple of minutes before the actual notification is triggered via the API.
6. Repeat step 4 and verify that the status has changed from 'Not confirmed' to 'Confirmed'.

7.9 Failed alternative payment notification

1. Repeat step 1-3 in [Successful alternative payment](#)
2. Verify that the customer is redirected to the alternative payment provider webpage. Verify the pending payment.
 - a) If you are testing against the test gateway a mock-up for bank and alternative payment solutions will be shown. If that is the case, click 'Developer options' and 'Open (Opens in a new window)'. Do not close the test bank page.
3. Verify that a new window appear in the browser and contain a summary page with correct information. Also verify that a message indicating that the payment has not been confirmed is shown. Take a note of the order number.
4. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly - 'Order Status' = Not confirmed.
5. Go back to the test bank page and click 'Call declined notification now'.
It can take a couple of minutes before the actual notification is triggered via the API.
6. Repeat step 4 and verify that the status has changed from 'Not confirmed' to 'Cancelled'.

7.10 Fraud checking (accepted credit card)

1. Repeat steps 1-6 in [Successful card payment](#)
2. Use a credit card number enabled for fraud checking and that returns the 'Accept' status.
 - a) For example: **4170000000000006**
3. Verify that the summary page is shown with the correct information and without any error messages. Take a note of the order number.
4. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly: 'Confirmation Status' = confirmed.
5. Login in to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Check that the amount corresponds with the information in Business manager and ensure that the status of the payment is 'preauth'.
6. Repeat with a credit card that returns the 'Challenge' status
 - a) For example: **5250000000000121**
7. Repeat with a credit card that returns the 'Unknown' status
 - a) For example: **5110000000000113**

7.11 Fraud checking (denied credit card)

1. Repeat steps 1-6 in [Successful card payment](#)
2. Use a credit card number enabled for fraud checking and that returns the 'Deny' status.
 - a) For example: **4170000000000105**

3. Verify that the user was redirected to the Billing page, and that the error message 'Card declined' is shown.
4. Go to: Merchant tools → Ordering → Orders in the Business manager. Locate and select the order and verify that the order has been handled correctly:
 - a) The status should be 'Failed'
 - b) Confirmation Status should be 'Not confirmed'
5. Login in to <https://testgateway.pensio.com> and locate the order by the order number via the search box in the top right corner. Check that the amount corresponds with the information in Business manager and ensure that the status of the payment is 'preauth'.

8 Operations, Maintenance

8.1 Data Storage

Please see 3.6 for what is saved outside Demandware.

8.2 Availability

If you experience any problems with the gateway or payments please contact AltaPay support. Please supply as much information as possible such as order/Payment ID, payment method, terminal name etc.

8.3 Support

If there is any problem with the payments or integration, please contact AltaPay support on support@altapay.com or +45 70 20 00 56 - option 1 (support).

9 User Guide

9.1 Roles, Responsibilities

The merchant must have access to AltaPay's test terminals before the integration can be completed. The merchant must set up terminals and user credentials correctly on 'Custom preference site' and perform tests on the test environment before going live for each shop.

AltaPay also need to verify that the test setup works prior to go-live. Please send customer access details to AltaPay support when the shop is ready for review.

9.2 Business Manager

No new features.

9.3 Storefront Functionality

Accept payments with different payment methods. For more information about what AltaPay can offer, please contact AltaPay at info@altapay.com.

10 Known Issues and limitations

Capture, release and refund is not available via Business Manager, but is accessible in fulfilment/order management. The merchant needs to implement this by him- or herself or perform these operations via the AltaPay Backend.

10.1 Planned changes

- Implement JS client library to make it easier for merchants to implement their own capture, refund and cancellation pipelines.
- Change error messages in the templates. This will make it easier for merchant to customize the error messages shown to the customer.
- Display decline/error reasons in Demandware backend.

These points are subjected to change.