

Paidy payment processing

Version<22.1.1>



Table of contents

Details

Table of contents	2
1. Summary	4
2. Component summary	5
2-1. Features overview	5
2-2. Restrictions and limitations	5
2.3. Use cases	6
2-3-1. Standard payments	6
2-3-2. Subscription payments	9
2-4. Privacy when paying	13
3. Implementation guide	14
3-1. Uploading cartridges (UX Studio)	14
3-2. BusinessManager setup	14
3-2-1. Cartridge assignment	14
3-2-2. Importing metadata	15
3-3. Modifying cartridges on the merchant	20
3-3-1. CheckoutJS installation	20
3-3-2. Controlling purchase confirmation buttons	22
3-3-3. Modifying resource files	23
3-4. Modifying cartridges on the merchant	23
3-4-1. Modifying COBillingjs	23
3-4-2. COPlaceOrder.js modifications	25
3-4-3. COSummary.js modifications	27
3-4-4. Modifying to package.json on the JS controller cartridge at the merchant	28
3-4-5. Modifying package.json on the int_paidy_controllers cartridge	28
3-5. Advice	28
4. Cartridge deployment testing at merchant	30
4-1. Before beginning a test	30
4-1-1. Paidy merchant management screen	30
4-1-2. Commerce Cloud Business Manager	31
4-2. PaidyCheckout test data	31
4-3. Browser compatibility	32
4-4. Test cases	32
4-4-1. Paidy standard payments	32
4-4-2. Paidy subscription payments	33
5. Webhook usage examples	34
5-1. Webhook usage	34
5-2. Detecting inconsistent payments	34
5-3. Handling inconsistencies	35
5-3-1. Processing example 1	35
5-3-2. Processing example 2	38
5-4. Webhook settings	38
6. What to do when the service is down	39
7. Revision history	39

1. Summary

Cartridges allow a Commerce Cloud store to make use of Paidy payments.

The developer must follow this text to install the cartridges and set them up on the online store.

Because Paidy allows for creating test payments by making use of a test API key, the user can test the settings before switching over to a live environment. For details on data needed for testing, see [4-1. Before beginning a test](#).

Three cartridges must be implemented in order to begin using Paidy payments.

- int_paidy is the core cartridge that enables use of Paidy payments on the store.
- int_paidy_controllers is a cartridge that enables support of Paidy payments on a JSController site.

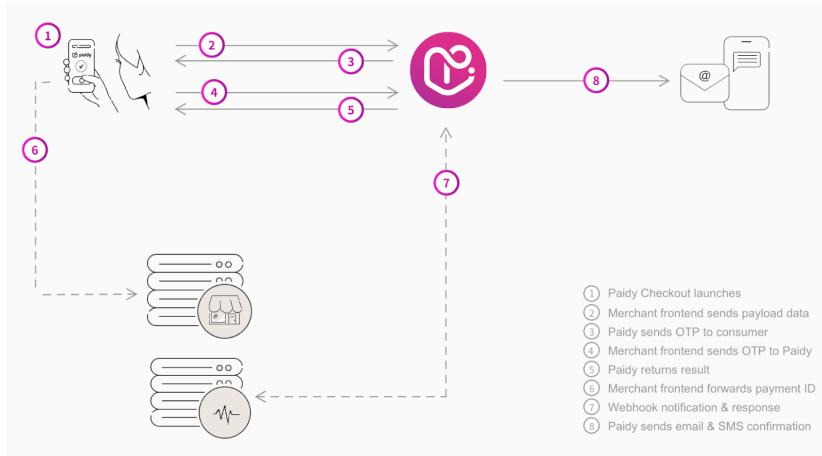
2. Component summary

2-1. Features overview

The process flow for Paidy is explained in the diagram below.

Paidy cartridges correspond to point 1-6 in this workflow.

In step 1-6, the results of authentication with Paidy are saved on the site.



2-2. Restrictions and limitations

- Only a “creation” process is in place for payment. Capture, Refund, Update, and Close must be implemented by the merchant.
- For first-time subscription payments, Commerce Cloud runs a creation process.
Subsequent payments must be configured at the merchant, but the payment process can be implemented by executing the functions found in the Paidy cartridge.
- Subscription payments are designed to only work when the user is logged in.
- Confirm notifications from the Webhook on OMS; if the payment process results are incorrect, you can choose to set the order status to shippable in SFCC, or cancel the payment on Paidy. For details, please see [5. Webhook usage examples](#).
- This cartridge can only be used on Japanese language sites. (language: Japanese; currency: JPY)

2.3. Use cases

2-3-1. Standard payments

Paidy Checkout launches when a user clicks a payment purchase button on a site.

Consumers enter the requisite information and make a Paidy payment.

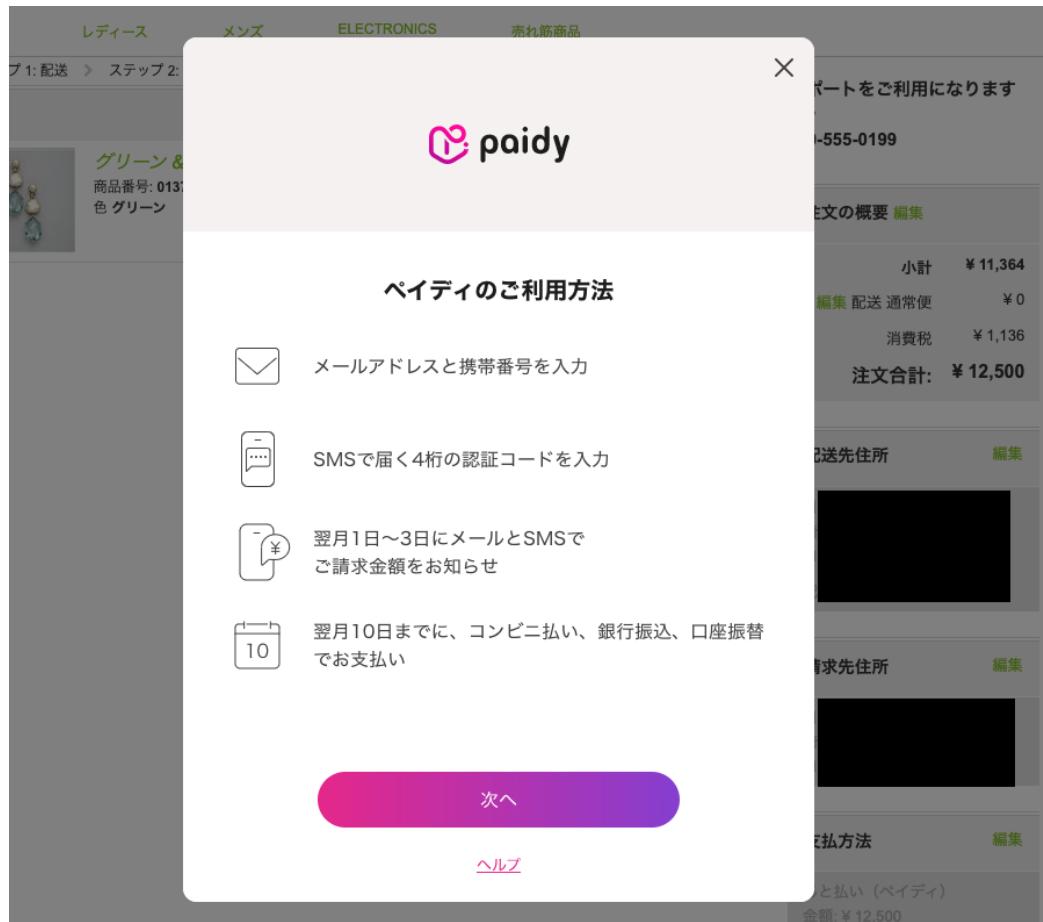
If the payment is successful, the process transitions to the product purchase confirmation screen.

- Payment flow

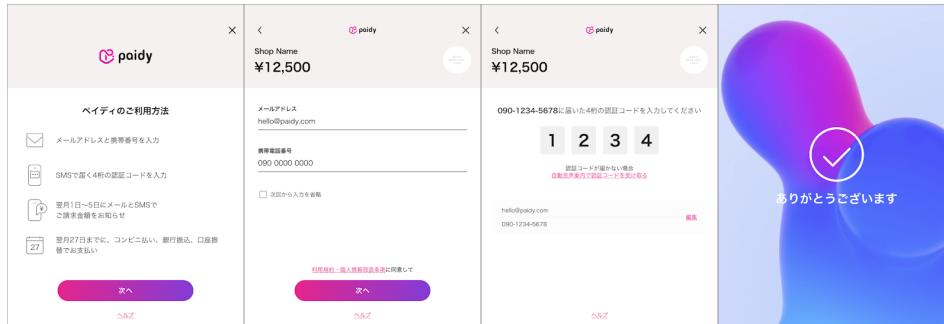
Select Paidy standard payments when selecting a payment method when checking out and proceed to confirming the order.



The Paidy Checkout popup appears.



The requisite information is filled in and Paidy payment is processed.



If the payment succeeds, the order is complete.

ご注文いただきありがとうございました

ご注文に関してご質問などがございましたら、月～金の午前8時～午後8時の間にお電話（800-555-0199）にてご連絡ください。

注文番号: 00000203

注文日: 18-10-2022

支护方法

文法力

金額・¥3,826

支払金額合計

小計 ￥3,61

配達通常便 ¥2

消音器 V 19

注意合計 Y 2,82

配送番号: 1

配送ステータス:	配送先
未配達	
配送方法:	
通常便	
商品	数量
ターコイズ & ゴールドフープイヤリング	1
配送追加料金	価格
商品番号:013742002799	¥3,617
色 ゴールド	

Order data created on Commerce Cloud

マーチャンツツール > 注文 > 注文 > 注文 WBSC00004505(Paidy_Master)

[全般](#) 属性 文仮 メモ 階層

注文 'WBSC00004505' の詳細

情報: 1件の配送先への1項目を含む。合計金額は¥ 10,436.00 です。

受取日: 10/26/17 9:04:07 AM Etc/UTC
サイト: Paidy_Master
作成者: 顧客
顧客: 山田 太郎
顧客ナンバー: 00029023
IP アドレス: 該当なし
Eメール: successful.payment@paidy.com
電話: 080-0000-0001

注文ステータス: 未送達 **確認ステータス:** 確認済み
配達ステータス: 未配達 **エクスポートステータス:** エクスポート可能

配送 00029024

個数	商品 ID	名前	Manufacturer (製造元)	税率	小売単価	税基準	項目合計
2	701642843610	Wide Waist Pencil Skirt 調整 金具ゼリ中		8.00 %	¥ 9,163.00	¥ 18,326.00 + ¥ 9,163.00	荷物追跡料 : ¥ 500.00

© 2017 salesforce.com, Inc. All Rights Reserved.

Paidy_Master タイムゾーン: 協定世界時 | インスタンスのタイムゾーン: 東部夏時間 | バージョン: 17.8 最終更新日時: 10月 18, 2017 (盈通モード: 15 分)
 取扱日付: 2017-10-18
 税金合計: ¥ 773.00
 合計: ¥ 10,436.00

[Eメールの送信](#) [注文の印刷](#)

Payment data created on Paidy

The screenshot shows the Paidy Management interface. On the left is a sidebar with icons for Home, Payment Management, Reports, and Settings. The main area has a header 'ホーム' (Home) and a sub-header '最近の支払' (Recent Payments). Below is a table with columns: 決済ID (Transaction ID), 顧客名 (Customer Name), 金額 (Amount), オーバー日時 (Overdue Date), ステータス (Status), and キャプチャ期限 (Capture Deadline). Six transactions are listed, all marked as 'AUTHORIZED'. The last transaction is highlighted with a red border.

If the payment fails, Paidy Checkout displays a payment error message and order failure data is created on Commerce Cloud.

Payment data will not be created on Paidy.

The four screenshots show the Paidy Checkout flow. Step 1: Enter shop name '¥12,500'. Step 2: Enter email 'hello@paidy.com' and phone '090-0000-0000'. Step 3: Enter verification code '090-1234-5678'. Step 4: An error message appears stating '今回のお支払いは承認されませんでした' (The payment was not approved) and '確認コードが間違っている場合、直接お支払い用QRコードを表示する' (If the verification code is incorrect, display the QR code for direct payment). The final step shows the same screen with the error message.

*Note that if the top-right X button is clicked on the popup, order failure data is created. Payment data will not be created on Paidy.

Order data created on Commerce Cloud (order failure)

The screenshot shows the '注文詳細' (Order Details) page for order 'WBSC00004503'. The status is '失敗' (Failed). The table below lists the order items and their statuses.

項目	値
注文ID	WBSC00004503
受取日	2017-10-26 22:22:22 UTC
サイト	Paidy_Master
販売者	顧客
顧客ナンバー:	[REDACTED]
IP アドレス:	該当なし
Eメール:	rejected.payment@paidy.com
電話:	333-333-3333
注文ステータス:	失敗
配送ステータス:	未配達
税金	8.00 %
小売価格	¥ 9,163.00
税率	¥ 732.64
項目合計	¥ 9,495.64
荷物配送料	¥ 500.00

At the bottom, there are buttons for 'Eメールの送信' (Send Email) and '注文の印刷' (Print Order).

2-3-2. Subscription payments

Only usable by logged in users.

When transitioning to a screen to confirm a purchase of products on the site, a check is performed to see if the consumer has a token.

When clicking on the order confirmation button for a first time purchase, there is no token, so Paidy Checkout launches. Consumers enter the requisite information and authenticate with Paidy. If the authentication completes correctly, a token is issued and is assigned to the consumer.

Once a token has been assigned or if the user has a token, the token data is used for Paidy payment.

- Payment flow

Select Paidy subscription payments when selecting a payment method when checking out and proceed to confirming the order.



An order confirmation screen is displayed and the order is confirmed.

The screenshot shows an order confirmation page from 'salesforce commerce cloud'. At the top, there's a navigation bar with 'commerce cloud', a search bar, and icons for location, user profile, and cart. Below the bar, it says '新規' (New), 'レディース' (Women), 'メンズ' (Men), 'ELECTRONICS', '売れ筋商品' (Best-Selling Products), and 'SPRING LOOK BOOK 2017'. The main content area shows a product summary: 'ワイドウエストペンシルスカート' (Wide-Waist Pencil Skirt) with details: '・全品セール中', '商品番号: 701642843610', '色 シンクマルチ', 'サイズ 4'. To the right is a '注文の概要' (Order Summary) table:

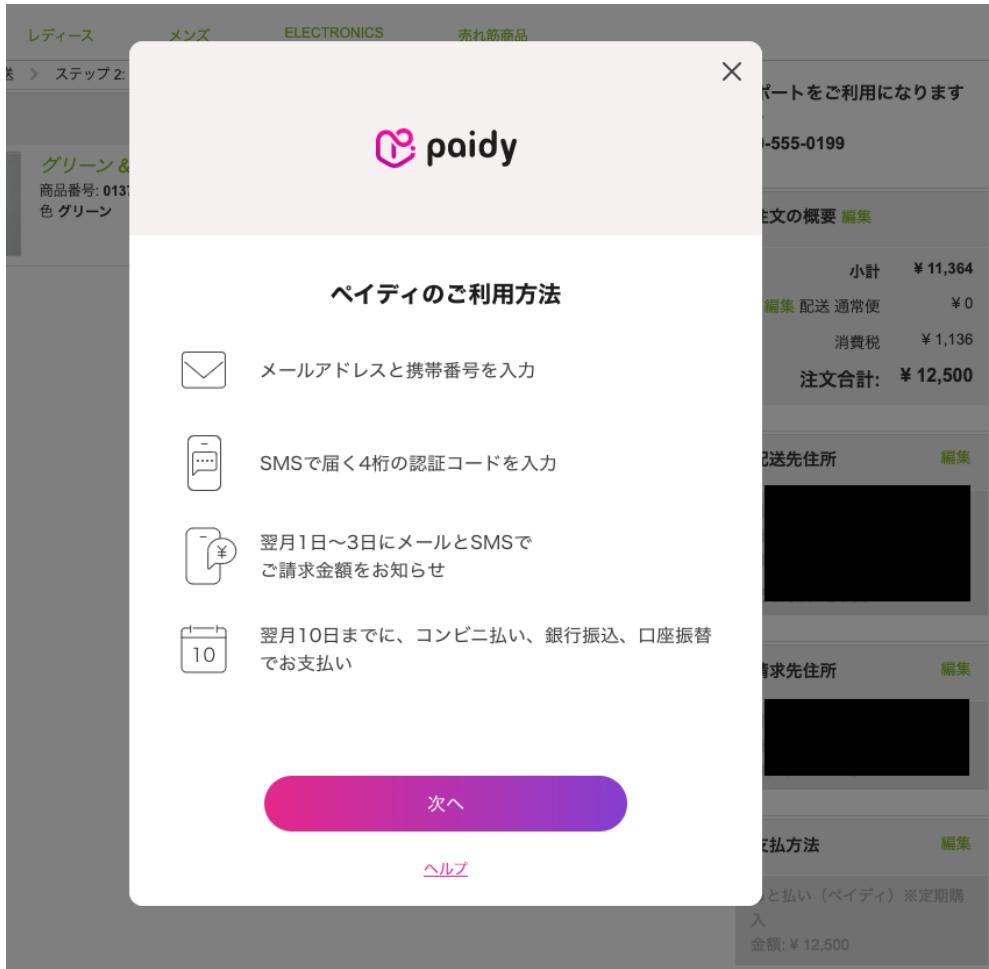
注文の概要	
小計	¥ 4,581
配送 通常配送	¥ 500
消費税	¥ 406
注文合計:	¥ 5,487

Below the summary is a breakdown of costs:

小計	¥ 4,581
配送 通常配送	¥ 500
消費税	¥ 406
注文合計:	¥ 5,487

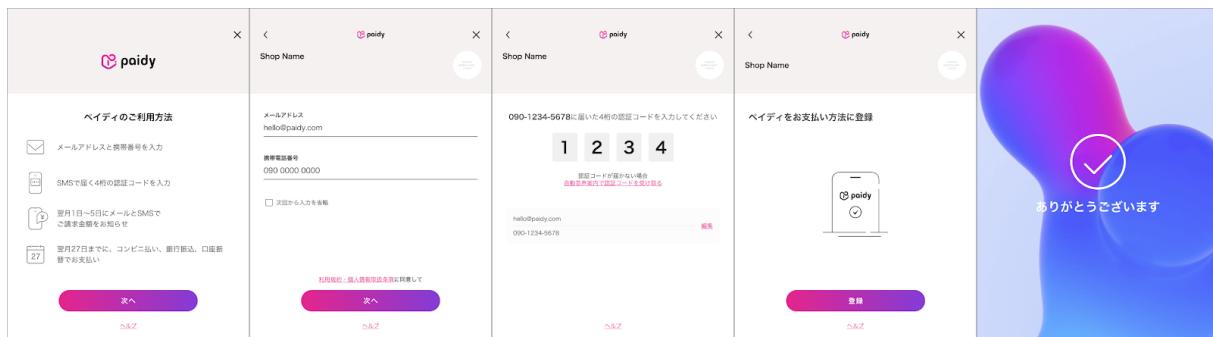
At the bottom, there are buttons for '買い物カゴを編集する' (Edit Shopping Cart) and '注文を確定する' (Confirm Order). To the right, there's a section for '配送先住所' (Delivery Address) with placeholder text '山田 太郎 ヤマダ タロウ' and '配送方法: 通常配送' (Delivery Method: Standard Delivery).

The Paidy Checkout popup appears (only the first time).



Enter the requisite information on Paidy Checkout and a token is issued.

*If the popup is closed during the process by the user clicking the X button on the top right, the popup will no longer display, but a token can be reissued and the payment processed.



After issuing a token, the process moves to payment processing. If the payment succeeds, we transition to the order confirmation screen.

ご注文いただきありがとうございました

ご注文に関してご質問などがございましたら、月～金の午前 8 時～午後 8 時の間にお電話 (800-555-0199) にてご連絡ください。

注文番号: 00000204

注文日: 18-10-2022

支払方法

あと払い(ペイディ)※定期購入
金額: ¥3,826

支払金額合計	
小計	¥ 3,617
配送 通常便	¥ 26
消費税	¥ 183
注文合計:	¥ 3,826

配送番号・1

配送ステータス:	配送先
未配達	
配送方法:	
通常便	
商品	
ターコイズ & ゴールドフープイヤリング	
配送追加料金	
商品番号: 013742002799	
色 ゴールド	
数量	価格
1	¥3,617

買い物に戻る

Order data created on Commerce Cloud

マーチャントツール > [注文](#) > [注文](#) > 注文: WBSC00004600(Paidy_Master)

全般 属性 文档 メモ 翻译

注文 'WBSC00004600' の詳細

Payment data created on Paidy

決済ID	顧客名	金額	オーリソリ日時	ステータス	キャプチャー期限	
pay_WfKlrF0AACwAkQfA TEST	山田 太郎 ヤマダ タロウ	¥5,487	2017-10-27 09:15:24	AUTHORIZED 取引ID: WBSC00004602	2017-11-26 10:15:24	-
pay_WfKxDx10AAEYAKqZN TEST	山田 太郎 ヤマダ タロウ	¥5,487	2017-10-27 09:54:31	AUTHORIZED 取引ID: WBSC00004600	2017-11-26 09:54:31	-
pay_WfGIyF0AAFOAkIc- TEST	山田 太郎 ヤマダ タロウ	¥10,436	2017-10-26 18:05:36	AUTHORIZED 取引ID: WBSC00004505	2017-11-25 18:05:39	-
pay_WfGSDFOAAE0AkK-s TEST	山田 太郎 ヤマダ タロウ	¥5,487	2017-10-26 16:43:08	AUTHORIZED 取引ID: WBSC00004502	2017-11-25 16:44:08	-
pay_WfGNjp0AAFMAMk1k TEST	山田 太郎 ヤマダ タロウ	¥5,487	2017-10-26 16:24:22	AUTHORIZED 取引ID: WBSC00004501	2017-11-25 16:24:25	-

If payment fails, a payment error will be displayed on the order confirmation page and order failure data is created on Commerce Cloud.

Payment data will not be created on Paidy.

Salesforce commerce cloud

キーワードまたは商品番号を入力

新着 レディース メンズ ELECTRONICS 売れ筋商品 SPRING LOOK BOOK 2017

ステップ 1: 記述 > ステップ 2: 請求 > ステップ 3: 注文の確定

ご注文を送信できませんでした。支払設定をご確認の上、再度試してください。お手数をおかけして申し訳ございません。

商品	数量	合計
ワイドウエストペンシルスカート ・新品セール中 商品番号: 701642843610 色 シンクマルチ サイズ 4	1 在庫あり	¥ 4,581

小計	¥ 4,581
配達 道常配達	¥ 500
消費税	¥ 406
注文合計:	¥ 5,487

配送先住所	編集
日本 888-8888 愛媛県 愛媛愛媛 愛媛住所 山田 太郎 ヤマダ タロウ	
配送方法: 道常配達	

« 買い物カゴを編集する **注文を確定する** 請求先住所 編集

Order data created on Commerce Cloud (order failure)

マーケティングツール > [注文] > [注文] > 注文: WBSC00004601(Paidy_Master)

全般 属性 支払 メモ 履歴

注文 'WBSC00004601' の詳細

情報:	1 件の配送先への 1 項目を含む。合計金額は ¥ 5,487.00 です。							
受取日:	10/27/17 1:06:16 AM Etc/UTC							
サイト:	Paidy_Master							
販売者:	顧客							
顧客:								
顧客ナンバー:								
IP アドレス:	該当なし							
Eメール:	rejected.payment@paidy.com							
電話:	080-0000-0001							
注文ステータス:	失敗	確認ステータス:	未確認	エクスポートステータス:	未エクスポート			
配送								
例数	商品 ID	名前	ステータス	Manufacturer (製造元)	税率	小売単価	税基準	項目合計
1	701642843610	ワイドウエストペンシルスカート ・調整 金合セール中	CANCELLED		8.00 %	¥ 9,163.00	¥ 9,163.00	¥ 0.00 ¥ 4,582.00 荷物配送料: CANCELLED ¥ 600.00

© 2017 salesforce.com, Inc. All Rights Reserved.

Paidy_Master タイムゾーン: 協定世界時 | インスタンスのタイムゾーン: 東部夏時間リージョン: 17.8 最終更新日時: 10月 18, 2017 (直換モード: 15.5)

税込合計:	¥ 5487.00
税金合計:	¥ 406.00
合計:	¥ 5,487.00

Eメールの送信 **注文の印刷**

2-4. Privacy when paying

Paidy payment authentication requires entering a telephone number and e-mail address, but these are not saved on Commerce Cloud.

However, information entered upon registration and shipment and billing details are saved on Commerce Cloud.

3. Implementation guide

3-1. Uploading cartridges (UX Studio)

The cartridge is uploaded to Commerce Cloud.

- int_paidy
- int_paidy_controllers (JSController sites)

The Paidy cartridge can be used on SiteGenesis-based JSController sites.

Use Commerce Cloud UX-studio to upload the int_paidy and int_paidy_controllers cartridges to Commerce Cloud.

3-2. BusinessManager setup

Paidy payment settings are configured on the Commerce Cloud management screen (Business Manager).

3-2-1. Cartridge assignment

A Paidy cartridge is assigned to the site.

Management > Sites > Site Management > Sites Under Management: Settings

管理 > サイト > サイトの管理 > SiteGenesis - 設定

全般 設定 キャッシュ サイトのステータス

SiteGenesis - 設定

適用をクリックして詳細を保存します。前回保存された状態に戻すには、リセットをクリックします。

インスタンスタイプ: Sandbox (サンドボックス)/Development (開発) ▾

廃止されています。HTTP と HTTPS のホスト名の設定では、サイトのエイリアス設定の新しい機能を使用することをお薦めします ("サイト URL エイリアス設定")。このセクションで設定する HTTP/HTTPS ホスト名の値は、エイリアスの設定でホスト名が定義されていない場合に、古い設定スタイルをサポートする目的でのみ使用されます。

HTTP ホスト名:

HTTPS ホスト名:

インスタンスタイプ: すべて

カートリッジ: init_paidy/sitegenesis_storefront_controllers:sitegenesis_storefront_core

有効なカートリッジのパス: sitegenesis_storefront_controllers:sitegenesis_storefront_core:plugin_apple_pay:plugin_facebook:plugin_pinterest_commerce:plugin_web_payments:core

<< リストに戻る



- JSController sites

int_paidy_controllers:int_paidy: merchant cartridge

Administration > Sites > Manage Sites > PaidyJDemo - Settings

General **Settings** Cache Site Status

PaidyJDemo - Settings

Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

Instance Type:	Sandbox/Development ▾
Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("Site URLs/Aliases Configuration"). The HTTP/HTTPS hostnames values set in this section will be used if no hostnames are defined by aliases configuration and are intended only to support an older configuration style.	
HTTP Hostname:	
HTTPS Hostname:	
Instance Type:	All
Cartridges:	<input type="text" value="int_paidy_controllers:int_paidy:paidy_storefront_controllers:paidy_storefront_core"/>
Effective Cartridge Path:	int_paidy_controllers int_paidy

3-2-2. Importing metadata

meta_data_22.1.0_sg.zip is provided in the package.

By importing this zip file, you can add services, add payment methods / payment processors, change system objects and custom objects to use Paidy payment. Details will be described later.

Administration > Site Development > Site Import & Export

Click [Choose File].

インポート

アーカイブのアップロード:

ローカル リモート

選択されていません

確認 変更 ▲

Select meta_data_22.1.0_sg.zip in the package and click the [Upload] button.

ファイルは、ローカルファイルである場合、またはファイルが作成されたインスタンスにログインしている場合にのみ削除できます。

アーカイブのアップロード:

ローカル リモート

meta_data...1.0_sg.zip

After uploading the zip file, check `meta_data_22.1.0_sg.zip` and click [Import].

The screenshot shows a file upload interface. At the top, there is a radio button for "ローカル" (Local) which is selected. Below that is a file selection input field containing "instance/meta_data_22.1.0_sg.zip". To the right of the input field is a "アップロード" (Upload) button. A table below lists files: "instance/meta_data_22.1.0_sg.zip" (Local, Yes, 3.48 KB, 6/10/22 2:49:15 AM). At the bottom right of the table is a red-bordered "インポート" (Import) button.

Click [OK].

インポート

選択したアーカイブをインポートしますか?

OK キャンセル

-Configuration when importing service metadata

Describes the service configuration when importing metadata.

Note that for services added after import

Please do not change the information described.

<Service Credentials>

Name: paidy.http.payment

URL: <https://api.paidy.com>

<Service Profiles>

Name: paidy.api.prof

<Services>

Name: paidy.api.payment

Type: HTTP

Enabled: check on

Service Mode: Live

Profile: paidy.api.prof

認証Credentials: paidy.http.payment

· Payment method · Configuration when importing payment processor metadata

Describes the payment method and payment processor configuration when metadata is imported.

As a point of caution, for payment methods and payment processors added after import

Please do not change the information described.

<Payment Processors >

ID: PAIDY_STANDARD

<Payment Processors >

ID: PAIDY_SUBSCRIPTION

<Payment Methods >

ID: PAIDY_STANDARD

Name: あと払い（ペイディ）

Enabled: Yes

<Payment Methods>

ID : PAIDY_SUBSCRIPTION

Name: あと払い（ペイディ）※定期購入

Enabled: Yes

- Object layout for metadata importing

Here we include instructions on the layout of objects when importing metadata.

One word of caution is that system and custom objects added after importing should not be modified.

<Order>

Management > Site Development > System Object Types > Order - Attribute Definitions

System objects: Order is an object that corresponds to order data

Order is given the below definitions as custom attributes.

paidyPaymentId: saves the ID issued when making a Paidy payment.

paidyToken: saves the token value used when using subscription payments.

Select All	ID	Name
<input type="checkbox"/>	paidyPaymentId	
<input type="checkbox"/>	paidyToken	

<Site Preferences>

Management > Site Development > System Object Types > Site Preferences > Attribute Definitions

System object: Site Preferences is an object that corresponds to site settings.

The below is defined as a custom attribute in Site Preferences.

paidy_api_key: public key data used to connect to the Paidy API.

This can be confirmed on the Paidy merchant management screen.

For a test, save the public key used for testing purposes.

paidy_enabled: Set to use or not use Paidy cartridges.

If turn it off, paidy payments will not be available as a payment method.

- Paidy merchant management screen (details found in section 4-1-1)

API

API version	2016-07-01
Production API key	
Public key	<input type="text" value="pk_live_"/>
Secret key	<input type="text" value="sk_live_"/>
Reissue public key	
Reissue secret key	
Test API key	
Public key	<input type="text" value="pk_test_"/>
Secret key	<input type="text" value="sk_test_"/>
Reissue public key	
Reissue secret key	

paidy_logo_url: the URL of a logo image to be displayed on the Paidy Checkout application.

If nothing is set, the Paidy logo will be shown.

paidy_secret_key: secret key data used to connect to the Paidy API.

This can be confirmed on the Paidy merchant management screen.

When testing, save the secret key used for testing purposes.

paidy_service_name: key data used to call the information for communicating with Paidy as registered in BM.

This is fixed to paidy.api.payment.

paidy_store_name: the shop name displayed on Paidy Checkout application headers, MyPaidy, and the merchant dashboard.

paidy_token_description (optional): specifies descriptions of tokens defined by the merchant.

<input type="checkbox"/>	<u>paidy_api_key</u>
<input type="checkbox"/>	<u>paidy_enabled</u>
<input type="checkbox"/>	<u>paidy_logo_url</u>
<input type="checkbox"/>	<u>paidy_secret_key</u>
<input type="checkbox"/>	<u>paidy_service_name</u>
<input type="checkbox"/>	<u>paidy_store_name</u>
<input type="checkbox"/>	<u>paidy_token_description</u>

<Customer Profile>

Management > Site Development > System Object Types > Customer Profile > Attribute Definitions

System object: Customer Profile is an object relating to customer data.

The below is defined as a custom attribute on the Customer Profile.

paidyToken: token data used for subscription payments. Saved when issuing a token.

Select All	ID	Name
<input type="checkbox"/>	paidyToken	

<Custom site environment settings>

Merchant Tools > Site Environment Settings > Custom Site Environment Settings Group

Once metadata is successfully imported, a group with the Paidy ID will be created in the custom environment settings group.

ID	Name	Description	Preferences	View Across Sites
Paidy	Paidy		6	View

Clicking Paidy displays configuration screens for each value. Enter the requisite information and click the Save button at the top right.

名前	値	デフォルト値
(paidy_enabled)	はい	はい
(paidy_api_key)	pk_test_	サイト全体で編集
(paidy_logo_url)	https://paidy.com/dist/paidy-logo-noshadow.svg	サイト全体で編集
(paidy_secret_key)	sk_test_	サイト全体で編集
(paidy_store_name)	PaidyDemoStore	サイト全体で編集
(paidy_service_name)	paidy.api.payment	サイト全体で編集
(paidy_token_description)		サイト全体で編集

3-3. Modifying cartridges on the merchant

3-3-1. CheckoutJS installation

In order to call Paidy Checkout and process payment, the Checkout JS script is embedded in the payment page template.

Including this script will call Paidy Checkout.

To embed the template, modify the following file.

- [storefront_core]\cartridge\templates\default\components\footer\footer_UI.isml

Add the below after <script src="\${URLUtils.staticURL('/js/app.js')}"></script>

```
.....  
<isscript>  
if (!empty(pdct.Basket)) {  
    var paymentMethods = pdct.Basket.getPaymentInstruments();  
    var paymentMethod = "";  
  
    if (paymentMethods != null && !paymentMethods.isEmpty() ) {  
        for( var i = 0; i < paymentMethods.length ; i++ ){  
            if( paymentMethods[i] != null && paymentMethods[i].paymentMethod != "GIFT_CERTIFICATE" )  
                paymentMethod = paymentMethods[i].paymentMethod;  
        }  
    }  
}  
</isscript>  
<isif condition="${paymentMethod=='PAIDY_STANDARD'}">  
<script type="text/javascript" src="https://apps.paidy.com/"></script>  
<script src="${URLUtils.staticURL('js/paidyStandard.js')}"></script>  
<iselseif condition="${paymentMethod=='PAIDY_SUBSCRIPTION'}">  
<script type="text/javascript" src="https://apps.paidy.com/"></script>  
<script src="${URLUtils.staticURL('js/paidySubscription.js')}"></script>  
<iselse/>  
<script>  
(function(){  
    var button = $('.submit-order button[type="submit"]');  
    if(button.length) {  
        button.prop('disabled', false);  
        button.removeClass('disabled');  
    }  
})();  
</script>  
</isif>  
.....
```

```

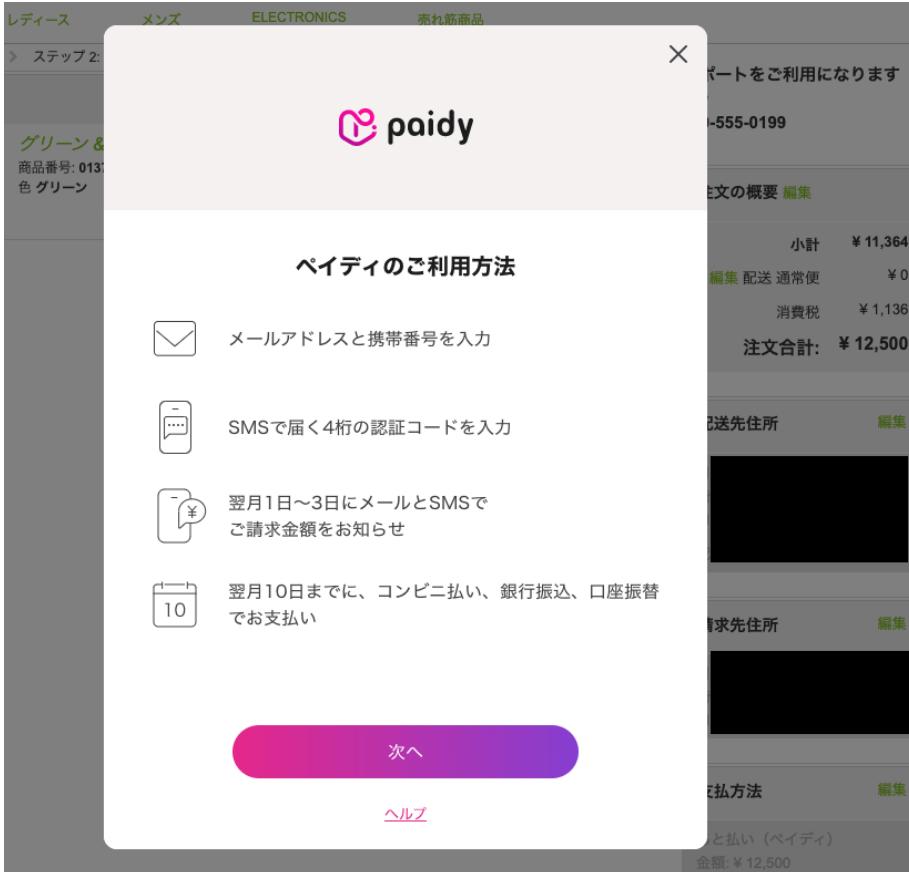
<script src="${URLUtils.staticURL('/js/app.js')}"></script>

<script>
if (!empty(pdct.Basket)) {
    var paymentMethods = pdct.Basket.getPaymentInstruments();
    var paymentMethod = '';

    if (paymentMethods != null && !paymentMethods.isEmpty() ) {
        for( var i = 0; i < paymentMethods.length ; i++ ){
            if( paymentMethods[i] != null && paymentMethods[i].paymentMethod != "GIFT_CERTIFICATE" )
                paymentMethod = paymentMethods[i].paymentMethod;
        }
    }
}
</script>
<isif condition="${paymentMethod=='PAIDY_STANDARD'}">
<script type="text/javascript" src="https://apps.paidy.com/"></script>
<script src="${URLUtils.staticURL('js/paidyStandard.js')}"></script>
<iselseif condition="${paymentMethod=='PAIDY_SUBSCRIPTION'}">
<script type="text/javascript" src="https://apps.paidy.com/"></script>
<script src="${URLUtils.staticURL('js/paidySubscription.js')}"></script>
<iselse/>
<script>
(function(){
    var button = $('.submit-order button[type="submit"]');
    if(button.length) {
        button.prop('disabled', false);
        button.removeClass('disabled');
    }
})();
</script>
</isif>

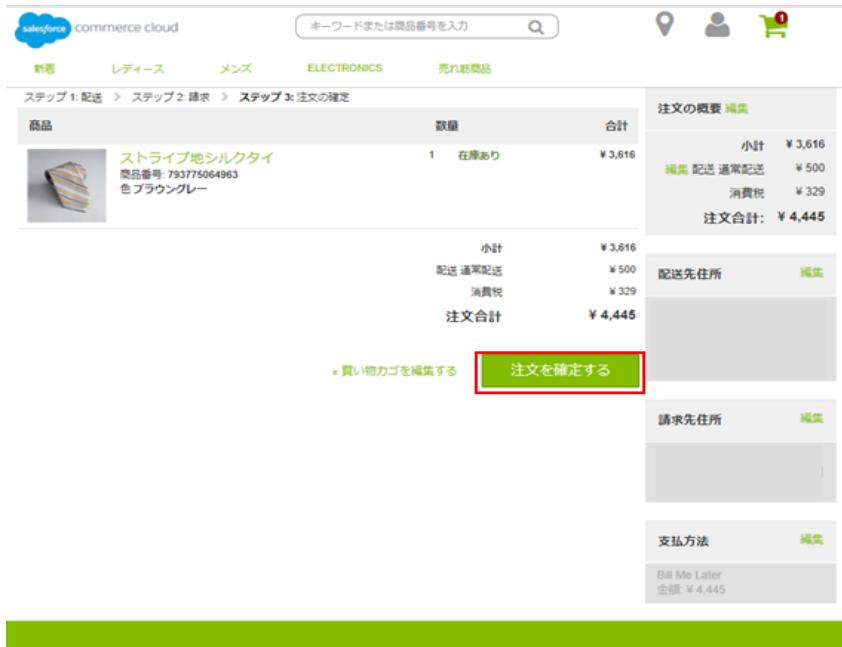
```

```
<isif condition="${!($('pageContext' in this) || empty(pageContext))}">
```



3-3-2. Controlling purchase confirmation buttons

When displaying the purchase confirmation screen and before the HTML source code is fully imported, if the confirm purchase button is clicked, the Paidy Checkout popup may fail to function properly.



Therefore, the purchase confirmation button must be styled such that it cannot be clicked when first displayed. The template implements this as follows.

- [storefront_core]\cartridge\templates\default\checkout\summary\summary.isml

A button with the button-fancy-large class is also given the disabled class and disabled attribute.

```
<button class="button-fancy-large disabled" type="submit" name="submit" value="${Resource.msg('global.submitorder','locale',null)}" disabled>
```

The tag is added immediately after the close tag on the `<form action="${URLUtils.https('COSummary-Submit')}" method="post" class="submit-order">` add tags immediately after the form tag.

```
<input type="hidden" name="selected_payment_method" value="${pdict.CurrentForms.billing.paymentMethods.selectedPaymentMethodID.value}" />
```

The above form tag also includes the submit-order class; this class should be left in and not deleted.

The purchase confirmation button will now only be clickable when Paidy Checkout is ready.

```
215 <form action="${URLUtils.https('COSummary-Submit')}" method="post" class="submit-order"> |  
216 |   <input type="hidden" name="selected_payment_method" value="${pdict.CurrentForms.billing.paymentMethods.selectedPaymentMethodID.value}" /> ↓  
217 |   <fieldset> ↓  
218 |     <div class="form-row"> ↓  
219 |       <a class="back-to-cart" href="${URLUtils.url('Cart-Show')}"/> ↓  
220 |       <input type="button" value="${Resource.msg('summary.editcart','checkout',null)}" encoding="off" /> ↓  
221 |     </div> ↓  
222 |     <button class="button-fancy-large disabled" type="submit" name="submit" value="${Resource.msg('global.submitorder','locale',null)}" disabled> ↓  
223 |       ${Resource.msg('global.submitorder','locale',null)} ↓  
224 |     </button> ↓  
225 |   </div> ↓  
226 |   <input type="hidden" name="${dw.web.CSRFProtection.getTokenName()}" value="${dw.web.CSRFProtection.generateToken()}" /> ↓  
227 | </fieldset> ↓  
228 | </form> |
```

3-3-3. Modifying resource files

Using Paidy payments requires including URL and timeout information in a configuration file called a resource file.

- [storefront_core]\cartridge\scripts\util\Resource.ds

Add the below at the end of the urls section of ResourceHelper.getUrls

```
.....  
getPaidyStandardPaidyConfig : URLUtils.https("PaidyStandard-GetPaidyConfig").toString(),  
doPaidyStandardCOSummarySubmit: URLUtils.https("COSummary-Submit").toString(),  
doPaidyStandardFailOrder : URLUtils.https("PaidyStandard-FailOrder").toString(),  
doPaidyStandardPlaceOrder : URLUtils.https("PaidyStandard-PlaceOrder").toString(),  
getPaidyStandardOrderToken : URLUtils.https("PaidyStandard-GetOrderToken").toString(),  
getPaidySubscriptionConfig : URLUtils.url('PaidySubscription-GetPaidyConfig').toString(),  
setPaidySubscriptionToken : URLUtils.url('PaidySubscription -SetPaidyToken').toString()  
.....
```

3-4. Modifying cartridges on the merchant

3-4-1. Modifying COBillingjs

A routine is added to clear the fields if the payment method is changed from credit card or BML to Paidy.

```
.....  
function resetPaymentForms() {  
  
    var cart = app.getModel('Cart').get();  
  
    var status = Transaction.wrap(function () {  
        if (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('PayPal')) {  
            app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();  
            app.getForm('billing').object.paymentMethods.bml.clearFormElement();  
  
            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));  
            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));  
            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));  
            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_SUBSCRIPTION'));  
        } else if  
(app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals(PaymentInstrument.METHOD_CREDIT_CARD)) {  
            app.getForm('billing').object.paymentMethods.bml.clearFormElement();  
  
            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));  
            cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));  
cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));  
    })  
}
```

```

        cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_SUBSCRIPTION'));
    } else if
(app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals(PaymentInstrument.METHOD_BML)) {
    app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();

    if (!app.getForm('billing').object.paymentMethods.bml.ssn.valid) {
        return false;
    }
}

cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
    cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));
    cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));
    cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_SUBSCRIPTION'));
} else if
(app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('PAIDY_STANDARD')) {
    app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();
    app.getForm('billing').object.paymentMethods.bml.clearFormElement();

    cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
        cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));
        cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));
        cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_SUBSCRIPTION'));
} else if
(app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('PAIDY_SUBSCRIPTION'))
{
    app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();
    app.getForm('billing').object.paymentMethods.bml.clearFormElement();

    cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
        cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));
        cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));
        cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));
}
return true;
});

return status;
}
.....

```

```

function resetPaymentForms() {
    var cart = app.getModel('Cart').get();

    var status = Transaction.wrap(function () {
        if (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('PayPal')) {
            app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();
            app.getForm('billing').object.paymentMethods.bml.clearFormElement();

            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));

            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_SUBSCRIPTION'));
        } else if (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals(PaymentInstrument.METHOD_CREDIT_CARD)) {
            app.getForm('billing').object.paymentMethods.bml.clearFormElement();

            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));

            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_SUBSCRIPTION'));
        } else if (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals(PaymentInstrument.METHOD_BML)) {
            app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();

            if (!app.getForm('billing').object.paymentMethods.bml.ssn.valid) {
                return false;
            }

            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));

            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_SUBSCRIPTION'));
        } else if (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('PAIDY_STANDARD')) {
            app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();
            app.getForm('billing').object.paymentMethods.bml.clearFormElement();

            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));
        } else if (app.getForm('billing').object.paymentMethods.selectedPaymentMethodID.value.equals('PAIDY_SUBSCRIPTION')) {
            app.getForm('billing').object.paymentMethods.creditCard.clearFormElement();
            app.getForm('billing').object.paymentMethods.bml.clearFormElement();

            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_CREDIT_CARD));
            cart.removePaymentInstruments(cart.getPaymentInstruments(PaymentInstrument.METHOD_BML));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PayPal'));
            cart.removePaymentInstruments(cart.getPaymentInstruments('PAIDY_STANDARD'));
        }
        return true;
    });
}

return status;
}

```

3-4-2. COPlaceOrder.js modifications

When creating an order in Ajax on Paidy, the below is added to suspend the process before executing payment.

```

.....
function handlePayments(order) {
(abbreviated)
} else {

var authorizationResult = PaymentProcessor.authorize(order, paymentInstrument);

if (authorizationResult.not_supported || authorizationResult.error) {
return {

```

```

error: true,
PlaceOrderError: authorizationResult.PlaceOrderError
};

}

}

}

}

return {};
}

function start() {
(abbreviated)
    var handlePaymentsResult = handlePayments(order);

    if (handlePaymentsResult.error) {
        return Transaction.wrap(function () {
            OrderMgr.failOrder(order);

if (handlePaymentsResult.PlaceOrderError != null) {
            return {
                error : true,
                PlaceOrderError : handlePaymentsResult.PlaceOrderError
            };
        } else {
            return {
                error : true,
                PlaceOrderError : new Status(Status.ERROR,'confirm.error.technical')
            };
        }
    });

} else if (handlePaymentsResult.missingPaymentInfo) {
(abbreviated)
}

if (request.httpParameterMap.format.stringValue == 'ajax') {
    return {
        Order: order,
        paidy_order_created: true
    };
}

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

```

```
    return orderPlacementStatus;
}
.....
```

3-4-3. COSummary.js modifications

When creating an order in Ajax on Paidy, the below is added to create a JSON response if the process is suspended before payment is processed.

```
function submit() {
  var placeOrderResult = app.getController('COPlaceOrder').Start();
  if (placeOrderResult.error) {
    start({
      PlaceOrderError: placeOrderResult.PlaceOrderError
    });
  } else if (placeOrderResult.order_created) {
    showConfirmation(placeOrderResult.Order);
  } else if (placeOrderResult.paidy_order_created) {
    var paidyCartridge = require('~/package.json').paidyCartridge;
    var currentForms = session.forms;
    app
      .getView({
        JSONResponse: require(paidyCartridge +
          '/cartridge/scripts/paidy/standard/authorize').getConfirmationPaidyJSON(
          currentForms.billing.paymentMethods.selectedPaymentMethodID.value,
          customer,
          placeOrderResult.Order
        )
      })
      .render('util/responsejson');
  }
}
```

```

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) {
        showConfirmation(placeOrderResult.Order);
    } else if (placeOrderResult.paidy_order_created) {
        var paidyCartridge = require('~/package.json').paidyCartridge;
        var currentForms = session.forms;

        app
            .getView({
                JSONResponse: require(paidyCartridge +
                    '/cartridge/scripts/paidy/standard/authorize').getConfirmationPaidyJSON(
                    currentForms.billing.paymentMethods.selectedPaymentMethodID.value,
                    customer,
                    placeOrderResult.Order
                )
            })
            .render('util/responsejson');
    }
}

```

3-4-4. Modifying to package.json on the JS controller cartridge at the merchant

Add a reference to the Paidy cartridge to the package.json used on the merchant's JSController cartridge.

```
.....  
{  
  "hooks": "./cartridge/scripts/hooks.json",  
  "paidyCartridge": "int_paidy"  
}  
.....
```

3-4-5. Modifying package.json on the int_paidy_controllers cartridge

Change the name of the cartridge called by “controllers” to the name of the JSController cartridge used by the merchant.

```
.....  
{  
  "paidyCartridge": "int_paidy",  
  "hooks": "./cartridge/scripts/hooks.json",  
  "controllers": "paidy_storefront_controllers"  
}.....
```

3-5. Advice

- Notes on use of subscription purchases

If you followed the steps to add a payment method as indicated in section 3-2-3, by default, you will have two separate entries -- standard payments and subscription payments.

支払方法の選択 • 必須

- あと払い (ペイディ) あと払い (ペイディ) ※
定期購入

If there are means to detect whether subscription payments have been selected when transitioning to the payment selection screen, CSS or other means can be used to restrict the display to only show Paidy next-month payment (convenience store/bank).

- Notes on tokens

Merchant Tools > Clients > Client Management > Client Numbers: All

Tokens can be checked via client data.

paidyToken:

If, for some reason, you want to reissue a token, enter a null value in paidyToken and have the consumer purchase another subscription payment.

- Order confirmation e-mails for subscription purchases

The int_paidy cartridge is designed such that order confirmation e-mails are sent in the same manner as standard e-mails even for subscription purchases. To customize the e-mail sent, you can modify the int_paidy cartridge.

- Error message when a subscription payment is rejected

Please refer to the following wording for the message when a subscription payment is rejected.

今回の決済は承認されませんでした

申し訳ございませんが、オンラインショップが提供する他の支払方法をご利用ください。

なお、審査結果の詳細につきましては開示できませんのであらかじめご了承ください。

Paidyに関するお問い合わせ

0120-971-918

4. Cartridge deployment testing at merchant

4-1. Before beginning a test

4-1-1. Paidy merchant management screen

- Confirming the test API key

Check the public and secret keys used for the test API; these will be entered into the custom site environment settings screen.

API

The screenshot shows a user interface for managing API keys. At the top, there is a field labeled "API version" with the value "2016-07-01". Below this, there are two sections: "Production API key" and "Test API key". Each section has two fields: "Public key" and "Secret key", each accompanied by a "Reissue" button.

Production API key	
Public key	<input type="text" value="pk_live_"/> <button>Reissue public key</button>
Secret key	<input type="text" value="sk_live_"/> <button>Reissue secret key</button>

Test API key	
Public key	<input type="text" value="pk_test_"/> <button>Reissue public key</button>
Secret key	<input type="text" value="sk_test_"/> <button>Reissue secret key</button>

4-1-2.Commerce Cloud Business Manager

- API key settings

From the above settings screen, enter the test API keys you found for paidy_api_key and paidy_secret_key in section 4-1-1. Paidy merchant management screen.

Name	Value	Default Value
paidy_api_key	pk_test_	
paidy_logo_url		Edit Across Sites
paidy_secret_key	sk_test_	Edit Across Sites
paidy_store_name	サンプル店	Edit Across Sites
paidy_time_out_second	300	Edit Across Sites
paidy_service_name	paidy.api.payment	Edit Across Sites

4-2.PaidyCheckout test data

- E-mail address and phone number

Shop Name ¥12,500		E-mail address and telephone usable for test cases
<p>メールアドレス successful.payment@paidy.com</p> <p>携帯電話番号 080 0000 0001</p> <p><input type="checkbox"/> 次回から入力を省略</p> <p>利用規約・個人情報収集処理に同意して</p> <p>次へ</p>		
Test cases	E-mail address	phone number
Success	successful.payment@paidy.com	080-0000-0001
Failure	rejected.payment@paidy.com	

*Note that the e-mail address and phone number for the above member data and shipping and billing address can differ.

- Authentication number



080-0000-0001に届いた4桁の認証コードを入力してください



認証コードが届かない場合
[自動音声案内で認証コードを受け取る](#)



4-3. Browser compatibility

This cartridge supports the following OSes and browsers.

Device - OS	Browser
PC: Windows	Edge: latest version
	Chrome: latest version
	Firefox: latest version
PC: MacOS	Chrome: latest version
	Safari: latest version
Smartphone: iOS	iPhone (latest version of iOS), Safari iPhone (latest version of iOS), Chrome
Smartphone: Android	Android 4.4 and later, Chrome

4-4. Test cases

Test cases to run after configuring the cartridge are listed below.

4-4-1. Paidy standard payments

Standard payment is tested using a guest and a member.

1. Successful order as guest

- If not logging in and selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a successful e-mail address and phone number will display the product purchase confirmation screen.
- The Commerce Cloud order number created on the Paidy merchant site should be displayed as a transaction ID
- The order status for the corresponding order number on Business Manager on Commerce Cloud will be new

2. Order success as member

- After logging in, when selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a successful e-mail address and phone number will display the product purchase confirmation screen.
- The Commerce Cloud order number created on the Paidy merchant site should be displayed as a transaction ID
- The order status for the corresponding order number on Business Manager on Commerce Cloud will be new

3. Failed order as guest

- If not logging in and selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a failing e-mail address and phone number will display an error message.
- The order status for the corresponding order number on Business Manager on Commerce Cloud will have failed

4. Order failure as member

- After logging in, when selecting a payment method for Paidy standard payments, the PaidyCheckout popup appears. Entering a failing e-mail address and phone number will display an error message.
- The order status for the corresponding order number on Business Manager on Commerce Cloud will have failed

4-4-2. Paidy subscription payments

Use of subscription payments always requires logging in. Once the user succeeds at authentication for a subscription payment, the paidyToken containing that member's data is saved. Subsequent purchases do not display the PaidyCheckout popup. For testing purposes, you may elect to delete the paidyToken from Business Manager on Commerce Cloud in order to display the popup again.

The screenshot shows the 'Manage Customers' interface. At the top, there are tabs for General, Addresses, Customer Groups, and Active Data. Below these, there's a section titled 'Profile Attributes' containing various fields like Salutation, Title, Company, Job Title, First Name, Last Name, Name Suffix, Gender, Birthday, Email, Next Birthday, and Second Name. The 'paidyToken' field is located at the bottom of this list, highlighted with a red dashed border. Above it, there's a note: 'paidyToken: []'.

1. Success with no token

- When logging in as a member with no token yet issued and selecting a payment method for Paidy subscription payments, the Confirm Order button is clicked and the PaidyCheckout popup appears. Entering a successful e-mail address and phone number and proceeding to processing the payment will display a product purchase confirmation screen
- The Commerce Cloud order number created on the Paidy merchant site should be displayed as a transaction ID
- The order status for the corresponding order number on Business Manager on Commerce Cloud will be new

2. Failure with no token

- When selecting a payment method for Paidy subscription payments and clicking the Confirm Order button, the PaidyCheckout popup appears. Entering a failing e-mail address and phone number and proceeding to processing the payment will display an error message
- The order status for the corresponding order number on Business Manager on Commerce Cloud will have failed

3. Order success with token

- When logging in as an account that has previously succeeded with a Paidy subscription payment and selecting a payment method for a Paidy subscription payment and clicking the Confirm Order button, a product purchase confirmation screen is displayed
- Because a token exists, the PaidyCheckout popup does not appear
- The Commerce Cloud order number created on the Paidy merchant site should be displayed as a transaction ID
- The order status for the corresponding order number on Business Manager on Commerce Cloud will be new

5. Webhook usage examples

5-1. Webhook usage

Paidy payments run credit holds for a purchase by means of communication directly between a consumer and Paidy over JavaScript. Therefore, one limitation of the system is that if the consumer bounces or the connection is cut, the e-commerce site may fail to obtain the results of payment confirmation and the stale payment data will remain on Paidy.

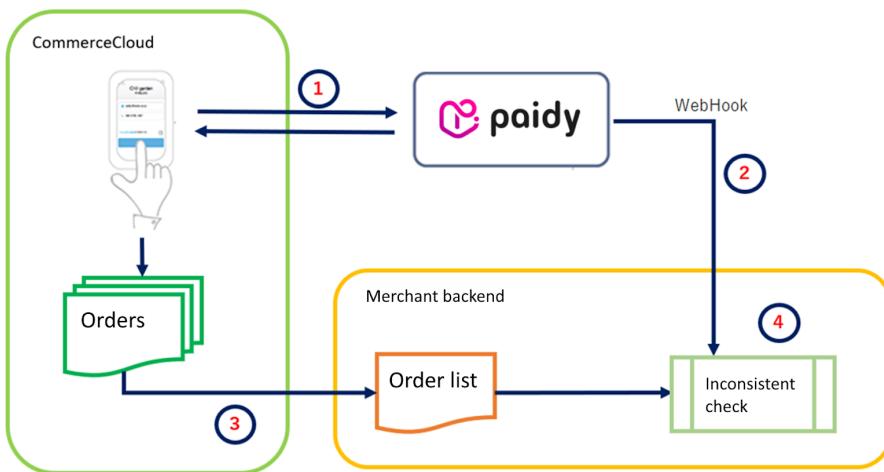
In the event of an inconsistency like the above, there are two problems.

1. The shop loses the opportunity to complete the order.
2. If the order does not have to be completed, the consumer has a credit hold placed against their Paidy account.

To accommodate these cases, a Webhook is provided.

The merchant can use the Webhook to detect inconsistent payments.

5-2. Detecting inconsistent payments



1. Payment

Payment is made via Commerce Cloud using Paidy. A callback from Paidy's payment results returns one of the three order statuses below.

- | | |
|---------------------------|--|
| - Payment success: New | Outputted to the order list. Payment possible status |
| - Payment failure: Failed | Not outputted to the order list. Payment not possible status |

- Processing: Created Not outputted to the order list. When in a payment not possible status, updating the payment transitions to a payable status.

2. Webhook

Setting Webhook endpoints on the merchant backend allows for receiving notifications from Paidy of payment completion.

3. Order list

A list of successful orders processed through batch processing on Commerce Cloud is sent as an xml file to the merchant's backend.

4. Inconsistent payment check processing

Checks for payments where Webhook data (2) is found, but no payment data (3) exists on Commerce Cloud.

5-3. Handling inconsistencies

Inconsistent payments that have been detected can be handled as follows, allowing for resolving the above issues.

5-3-1. Processing example 1

In this solution, a list of inconsistent/invalid payments is outputted and can be checked by the merchant. After a check is performed, the necessary steps are taken (approving or cancelling payment).

1. Establishing payment

Use the Order function on Business Manager on Commerce Cloud to update the status.

- General tab

Merchant Tools > Ordering > Orders > Order: WBSC00005100(Paidy_Master)

General Attributes Payment Notes History

Details for Order 'WBSC00005100'

Information:	Contains 1 line item to 1 shipping location The total price is ¥14,506.00.		
Date Received:	11/24/17 6:27:23 am Etc/UTC		
Site:	Paidy_Master		
Created By:	Customer		
Customer:	[REDACTED]		
Customer No.:	00033017		
IP Address:	n/a		
Email:	[REDACTED]		
Phone:	[REDACTED]		
Order Status:	Open	Confirmation Status:	Confirmed
Shipping Status:	Not Shipped	Export Status:	Ready for Export

The status of the above three points is changed per below

Order Status: Created→Open

Confirmation Status: Not Confirmed→Confirmed

Export Status:Not Exported→Ready for Export

- Attributes tab

The screenshot shows the Salesforce Merchant Tools interface for managing an order. The top navigation bar includes links for Sandbox, Merchant Tools, Administration, and Store. The current page path is Merchant Tools > Ordering > Orders > Order: WBSC00005100(Paidy_Master). The tab bar at the top has tabs for General, Attributes (which is selected), Payment, Notes, and History.

The main content area is titled "Attributes for Order 'WBSC00005100'". A note below the title states: "On this page you can edit the attributes of the order. Fields with a red asterisk (*) are mandatory. Click Apply to save changes. Click Revert to undo changes." The form contains several input fields:

- deliveryDate: [Input field]
- deliveryTimeSpan: [Input field]
- paymentCode: [Input field]
- Channel Type: [- None -] [Dropdown]
- Replacement Order Number: [Input field]
- Cancel Code: [- None -] [Dropdown]
- Shipping Status: * [0 (NOTSHIPPED)] [Dropdown]
- Customer Order Reference: [Section header]
 - Replaced Order No (replacedOrderNo): [Input field] (highlighted with a red box)
 - Replaced Order No: [Input field]
 - Replace Code: [- None -] [Dropdown]
 - External Order Status: [Input field]
 - Payment Status: * [0 (NOTPAID)] [Dropdown]
- PaidyPayments:
 - paidyToken: [Input field] (highlighted with a red box)
 - paidyPaymentId: [Input field] (highlighted with a red box) containing the value "pay_20171124_001"

Set the paidyPaymentID that arrived via the Webhook for paidyPaymentId

*In order to e-mail an order confirmation e-mail as you would with StoreFront, it must be customized, as the format differs (see below).

- Order confirmation e-mail (concept)

■SiteGenesis StoreFront e-mail (normally used e-mail)



The screenshot shows an order confirmation email from SiteGenesis. At the top, there are two sections: "Salesforce Commerce Cloud SiteGenesis" on the left and "Salesforce Commerce Cloud SiteGenesis" with address details (5 Wall Street, Burlington, MA 01803, salesforce.com, 電話:+1.800.555.0199) on the right. Below this is a grey header bar containing the text "ご注文いただきありがとうございました". The main body of the email contains a message in Japanese: "ご注文に関してご質問などがございましたら、お電話 (012-345-XXXX) にてご遠慮なくお問い合わせください (月曜日～金曜日、午前 8 時～午後 8 時)。". At the bottom, there is order information: "注文日: 24-11-2017" and "注文番号: WBSC00005102". A "支払情報" (Payment Information) section is shown, which includes a redacted "請求先住所" (Billing Address), "支払方法" (Payment Method: あと払い (ペイディ)), "金額: ¥ 2,492", and "支払金額合計" (Total Payment Amount: 小計 ¥ 1,808, 配送 通常配達 ¥ 500).

E-mails sent from SendMail button on bottom of General tab on BusinessManager > Orders

SiteGenesis

Demo SiteGenesis
5 Wall Street
Burlington, MA 01803
[SiteGenesis](#)
Phone: +1.888.553.9216

Order Confirmation: Order Number: WBSC00005101 - Thank you for your order!

This email confirms we received your order at [SiteGenesis](#). We will email your shipping confirmation with your tracking number when your items have been shipped. Your order will generally be shipped within 24 hours after we've received it. You can check your order's status on your account pages at [SiteGenesis](#)

Thank you for shopping with us!

Please print and save a copy for reference.

Please review the order below and verify that everything is correct.

PLEASE NOTE: Since we begin processing your order shortly after you submit it on our web site, if any changes are necessary you MUST contact us via PHONE ONLY, at **+1.888.553.9216**

HIPMENT 1

PRODUCT DETAILS	QTY.	TOTAL	SHIPPING DETAILS
Samsung Series 6 22" LCD High Definition Television Item Number:samsung-ln22a650 全品セール中	1	¥ 66,318 ¥ - 33,159	

PAYMENT METHOD	AMOUNT
Payment Total	¥ 36,351

2. Cancelling payments

Displaying the transaction in question on the Paidy merchant management screen and Canceling it (do not capture and close).

支払詳細

ステータス	AUTHORIZED
取引ID	WBSC00005640
Authorized金額	¥ 5,487 (注文金額: ¥4,581 税金: ¥406 配送: ¥500)
キャプチャー金額	¥ 0 (注文金額: ¥0 税金: ¥0 配送: ¥0 リファンド: ¥0)
日付	2017-12-01 14:44:40
キャプチャー期限	2017-12-31 14:44:41
ストア名	[REDACTED]
決済ID	pay_WiDsSEoAAFYAdAYR TEST

顧客情報

氏名	フリガナ	メールアドレス	住所
山田 太郎	ヤマダ タロウ	successful.payment@paidy.com	[REDACTED]

オーダー内容

商品番号	商品名	価格	数量	小計
701642843610	ワイドウエストパンツルスカート	¥4,581	1	¥4,581

5-3-2. Processing example 2

In this example, the system makes a determination and automatically cancels inconsistent payments. If, after referencing the order data and Webhook data on Commerce Cloud, the payment does not complete on Commerce Cloud, and authorization data is found only on Paidy, the close API is requested and the payment is closed.

<https://paidy.com/docs/jp/payments.html#close>

5-4. Webhook settings

In order to receive notifications from Paidy on the backend, the Webhook URL must be saved to the Paidy merchant management screen settings.

URL: <https://merchant.paidy.com>

You can enable these settings for a live environment by setting the merchant's live backend URL in the Webhooks: Live API Key and Old Version API Access Key sections.

In order to check connectivity with test payments before configuring the live environment, set the merchant's backend test URL in Webhooks: Test API Key.

Paidy merchant management screen

Webhooks - "production API key" and "old version API access key"

Webhook is not set up

Webhook URL

Currently configured Webhook URL Configuration

Webhooks - "Test API Key"

Setting for test Webhook (* If you turn it off, the currently set Webhook URL will be deleted)

Test Webhook URL

Currently set test webhook URL

6. What to do when the service is down

- what happens when the service is down

If you select Paidy payment, you will get an error screen or the pop-up will stop working.

- what the merchant should do to fix it

To get started, please contact Paidy Technical Support.

Paidy Technical Support

Mail: tech-support@paidy.com

TEL : 03-5545-5099 (weekdays 10:00-18:00)

If it takes a time to recover, please set the enable flag for paidy_enabled to "いいえ(No)" in the Custom Preferences.

インスタンススタイル
Sandbox (サンドボックス)

IDで検索... / 7

名前	値	デフォルト値
paidy_enabled (paidy_enabled)	いいえ	はい

サイト全体で編集

7. Revision history

Version	Date	Revisions
17.10.0	2017/10/13	First edition

18.01.0	2018/01/09	<ul style="list-style-type: none"> - Revised steps to account for developing cartridges with JSController and use on both Pipeline sites and JSController sites - Revisions pursuant to deprecated Webhook on Commerce Cloud Added 5. Operational examples
19.01.0	2019/10/25	<ul style="list-style-type: none"> -Correct meta information file and update usage procedure. -The function "ServiceRegistry" used in JScontroller is abolished and "LocalServiceRegistry" is used. -Implement "getResponseLogMessage" in the service process called by JScontroller. -All HttpClient calls now support service usage. -Add DocComment to each function.
20.1.0	2020/01/15	<p>3-2-2. Importing metadata</p> <ul style="list-style-type: none"> -Modify 「meta_data_19.1.0.zip」 to 「meta_data_20.1.0.zip」
20.1.0	2020/4/5	<ul style="list-style-type: none"> - The following changes were made due to the file name change <p>3-3-1. CheckoutJS installation</p> <p>before :</p> <pre><script src="\${URLUtils.staticURL('js/paidy_ normal.js')}"></script></pre> <pre><script src="\${URLUtils.staticURL('js/paidy_ regular.js')}"></script></pre> <p>after :</p> <pre><script src="\${URLUtils.staticURL('js/paidy Normal.js')}"></script></pre> <pre><script src="\${URLUtils.staticURL('js/paidy Regular.js')}"></script></pre> <ul style="list-style-type: none"> -Add 6. What to do when the service is down - Removed references to the Pipeline site.
21.1.0	2021/6/21	-Update 1. Cartridges version update

		-Update 15. Changes: Changed "meta_data_20.1.0.zip" to "meta_data_21.1.0_sg.zip"
22.1.0	2022/6/21	<p>p.1 update version in cover</p> <p>all pages Modified normal payment ID and subscription ID, as follows.</p> <ul style="list-style-type: none"> - normal -> standard - regular -> subscription <p>p.4 updated supported store Front Reference Architecture and compatibility mode version</p> <p>p.17 update import meta file name</p> <p>p.37 4-3. Removed IE11 from Browser compatibility</p>
22.1.1	2022/11/24	<p>All pages The following modifications were made due to the change in the name of Paidy standard and subscription payment service.</p> <ul style="list-style-type: none"> ● standard payment あと払い(ペイディ) ● subscription payment あと払い(ペイディ)※定期購入 ● 支払方法説明 <ul style="list-style-type: none"> ・クレジットカード、事前登録不要。 ・メールアドレスと携帯番号だけで、今すぐお買い物。 ・1か月に何度もお買い物しても、お支払いは翌月まとめて1回でOK。 ・お支払いは翌月10日までに、コンビニ払い・銀行振込・口座振替で。

		<p>ペイディについて詳しくはこち ら。</p> <p>Capture was changed to reflect the above changes.</p>
--	--	---