

Worldline-Direct Payments Cartridge

Worldline

Exported on 06/11/2024

Table of Contents

1	Functionality Overview.....	6
2	Install the cartridge	7
2.1	Before you begin	7
2.2	Step 1: Install the cartridge	7
2.3	Step 2: Install Node modules	8
2.4	Step 3: Build the code	8
2.5	Step 4: Add System Integrator Information (optional)	8
2.6	Step 5: Upload the code.....	8
2.7	Step 6: Import the metadata	8
3	Set up the cartridge.....	10
3.1	Before you begin	10
3.2	Step 1: Set up the cartridge path.....	10
3.2.1	Set up the site cartridge path in the Business Manager:.....	10
3.2.2	Set up the Business Manager cartridge path in the Business Manager:	10
3.3	Step 2: Set up Business Manager permissions in the Business Manager	11
3.4	Step 3: Connect your Commerce Cloud store to Worldline-Direct API.....	11
3.4.1	Step 3.1: Configure the service credentials	11
3.4.2	Step 3.2: Configure the service	12
3.5	Step 4: Set up Worldline-Direct Payments Cartridge	12
3.5.1	Step 4.1: Set up Worldline-Direct Payments Cartridge Configuration	12
3.5.2	Step 4.2: Configure Hosted Checkout Page Settings	13
3.5.3	Step 4.3: Configure Hosted Tokenization Page Settings	14
3.5.4	Step 4.4: Configure Intersolve payment ID (optional)	14
3.6	Step 5: Verify your connection to Worldline-Direct API	14
4	Take payments with the cartridge	15
4.1	Payment methods.....	15
4.2	Taking payments.....	15

4.2.1	Hosted Checkout Page.....	16
4.2.2	Hosted Tokenization Page.....	16
4.2.3	Sending line item prices to the API.....	16
4.3	PCI Compliance.....	17
5	Manage payments with the cartridge.....	18
5.1	Open Details Dialog for a transaction in the Business Manager	18
5.2	Capture transactions in the Business Manager.....	18
5.2.1	View all captures for a transaction.....	19
5.2.2	Manual capture a transaction	20
5.2.3	Automatic capture a transaction	20
5.3	Refund transactions in the Business Manager	20
5.3.1	View all refunds for a transaction.....	21
5.3.2	Refund a transaction	21
5.4	Cancel transaction within the Business Manager	22
6	Receive payment updates - Webhooks	23
6.1	Configure Salesforce Commerce Cloud to receive payment updates	23
6.1.1	Step 1: Set up webhooks in Worldline-Direct Backoffice and Salesforce Commerce Cloud	23
6.1.2	Step 2: Set up the Process Webhooks Job	23
6.2	Webhook handling and processing	24
6.2.1	Webhook event process flow	25
6.2.2	Business processes triggered by Process Webhooks Job	25
6.2.3	Webhook events and process mapping	26
6.2.4	Logging and debugging.....	28
7	Test the cartridge.....	29
7.1	Test cases (Registered + saved cards/Guest user HCP).....	29
7.2	Test cases (Registered + saved cards/Guest user HTP).....	31
8	Appendix 1: Salesforce Commerce Cloud data model extensions	36
8.1	System Objects	36

8.1.1	Site Preferences	36
8.1.2	Order	37
8.1.3	Customer Payment Instrument.....	38
8.1.4	Order Payment Instrument	38
8.2	Custom Objects	39
8.2.1	WorldlineDirectWebhooks	39
8.2.2	WorldlineDirectPaymentProducts.....	40
9	Appendix 2: Scheduled Jobs	41
9.1	Check Payment Status Job	42
9.1.1	Description	42
9.1.1.1	Processing orders in status “CREATED” which have a successful payment.....	42
9.1.1.2	Processing unconfirmed orders	42
9.1.2	Configuration.....	42
9.1.3	Schedule	43
9.2	Automatic Capture Job	43
9.2.1	Description	43
9.2.2	Configuration.....	43
9.2.2.1	Capture payments, placed more than X days ago	44
9.2.2.2	Capture all pending payments once per day, at the end of the day.....	44
9.2.3	Schedule	45
10	Appendix 3: Business processes	46
10.1	Validate Payment Business Process	46
10.1.1	Description	46
10.1.2	Steps	46
10.1.3	Flowchart.....	47
10.1.4	Worldline-Direct Payment statuses mapping.....	49
10.2	Process Unconfirmed Order Business Process.....	50
10.2.1	Description	50

10.2.2	Steps	50
10.2.3	Flowchart.....	51
10.2.4	Worldline-Direct API Payment statuses mapping.....	51
10.3	Handle Capture Business Process	52
10.3.1	Description	52
10.3.2	Steps	53
10.3.3	Flowchart.....	54
10.3.4	Worldline-Direct API Capture statuses mapping	54
11	Appendix 4: Services.....	56
12	Appendix 5: Failover and Recovery	57
12.1	Log files	57
13	Appendix 6: Upgrading the cartridge	58
13.1	Deleting version 21.* code	58
13.2	Deleting version 21.* metadata.....	58
13.3	Importing the new version.....	58

1 Functionality Overview

This cartridge provides integration of Worldline's payment system into a Salesforce Commerce Cloud store. It's built on top of SFRA v6.3.0 and when set up it will display Worldline's payment methods on the Salesforce checkout billing page and allow the payment to be completed and processed using them.

The merchant needs to have an account in Worldline's Backoffice in order to configure the communication between Salesforce and Worldline.


The process of setting up and configuring the integration is explained in the following sections.

2 Install the cartridge

2.1 Before you begin

The cartridge has been tested on the following platform versions:

- Salesforce Commerce Cloud platform Compatibility Modes 21.2, 21.7, 22.1 and 22.7
- SFRA version 6.0.0, 6.1.0, 6.2.0, 6.3.0
- All locales should work properly, as long as there are payment methods enabled for the specific country in the Worldline Backoffice

 The cartridge can be used with other SFRA/platform versions but it may require code changes performed by you or your system integrator.


 The cartridge is not compatible with SiteGenesis versions using Pipelines or Controllers.

2.2 Step 1: Install the cartridge

Install the cartridge to your project root directory at the same level as **storefront-reference-architecture**.

Example folder structure:

```
my-project/
|-- link_worldline_direct/
|   |-- cartridges/
|   |   |-- bm_worldline_direct/
|   |   |-- int_worldline_direct/
|   |-- documentation/
|   |-- metadata/
|-- storefront-reference-architecture
|   |-- cartridges/
|   |   |-- app_storefront_base/
|   |   |-- modules/
```

 If you have renamed the **storefront-reference-architecture** folder, please make sure to update the **base** path in package.json

2.3 Step 2: Install Node modules

The recommended Node version to use is 12.21.0 - this is also the recommended version for SFRA v6. From the cartridge's root directory, install Node modules using your command line:

```
npm install
```

2.4 Step 3: Build the code

From the cartridge's root directory, compile the client-side assets using your command line:

```
npm run build
```

2.5 Step 4: Add System Integrator Information (optional)

If your integration with Worldline-Direct API is built by a system integrator let us know their name, so that we can offer you better support.

1. Open **int_worldline_direct/cartridge/templates/resources/worldlineDirectSDK.properties**
2. Replace **YOUR_COMPANY_NAME** with the system integrator's company name.

```
worldline.direct.sdk.integrator=YOUR_COMPANY_NAME
```

2.6 Step 5: Upload the code

Upload **int_worldline_direct** and **bm_worldline_direct** using Commerce Cloud UX Studio or sgmf-scripts command line utils.

2.7 Step 6: Import the metadata

To add new configuration items, import the predefined metadata:

1. Open the **/metadata/site_import/sites/** folder.
2. Rename the **yourSiteId** folder to the ID of your site in the Business Manager.
3. Zip the **site_import** folder.
4. In the Business Manager, go to **Administration > Site Development > Site Import & Export** and import the zipped file.

After the import, attributes named **worldlineDirect[attributeName]** are added to:

- **Administration > Site Development > System Object Types > Site Preferences > Attribute Definitions**
- **Administration > Site Development > System Object Types > Order > Attribute Definitions**
- **Administration > Site Development > System Object Types > OrderPaymentInstrument > Attribute Definitions**
- **Administration > Site Development > System Object Types > CustomerPaymentInstrument > Attribute Definitions**
- **Administration > Site Development > Custom Object Types**

Also, the following service is added to **Administration > Operations > Services**:

- **worldline.https.direct.yourSiteId**

3 Set up the cartridge

3.1 Before you begin

Before you begin, make sure that you have performed the following steps:

1. Create an active [test/live account](#) with Worldline-Direct.
2. Install the cartridge.
3. In Salesforce Commerce Cloud, [enforce HTTPS](#)¹.



To be compatible with the Chrome v80 Cookie policy, you need to enforce the use of HTTPS for all sites.

3.2 Step 1: Set up the cartridge path

3.2.1 Set up the site cartridge path in the Business Manager:

1. Go to **Administration > Sites > Manage Sites > [yourSite] > Settings**.
2. In the **Cartridges** box add **int_worldline_direct** in front of **app_storefront_base**
Example cartridges configuration:

```
int_worldline_direct:app_storefront_base
```

3. Select **Apply**.

3.2.2 Set up the Business Manager cartridge path in the Business Manager:

1. Go to **Administration > Sites > Manage Sites > Manage the Business Manager site > Settings**

¹ https://documentation.b2c.commercecloud.salesforce.com/DOC2/topic/com.demandware.dochelp/content/b2c_commerce/topics/admin/b2c_enforce_https.html?resultof=%22%65%6e%66%6f%72%63%65%22%20%22%65%6e%66%6f%72%63%22%20%22%68%74%74%70%73%22%20%22%68%74%74%70%22%20

2. In the **Cartridges** box add **bm_worldline_direct:int_worldline_direct** in front of any other cartridges.
Example cartridges configuration:

```
bm_worldline_direct:int_worldline_direct:bm_app_storefront_base:bm_custom_plugin
```

3. Select **Apply**.

3.3 Step 2: Set up Business Manager permissions in the Business Manager

1. Go to **Administration > Organization > Roles & Permissions > [roleToUpdate] > Business Manager Modules**
2. From the pop-up select all storefront sites that will use Worldline-Direct Payments Cartridge
3. Find Ordering modules in the modules table
4. Assign the **Worldline-Direct Transactions** module permissions for the selected role.

3.4 Step 3: Connect your Commerce Cloud store to Worldline-Direct API

To connect your Commerce Cloud store to the Worldline-Direct API, you need to set up an API key in the Worldline-Direct Backoffice, and copy it over to the Business Manager.

Information on how to set up an API key in Worldline-Direct Backoffice can be found here. [Read our dedicated guide](#)² to learn how to generate one.

To copy the API key over to the Business Manager, go to **Administration > Operations > Services**

3.4.1 Step 3.1: Configure the service credentials

1. Open the Credentials tab
2. Open **worldline.https.direct.yourSiteId.TEST** and edit:
 - a. Name: replace **yourSiteId** with the actual ID for your site
 - b. User: enter your API key ID.
 - c. Password: enter your API key secret.
3. Select **Apply** in the bottom-right corner of the screen.

² <https://support.direct.ingenico.com/documentation/api/authentication>



Make sure you don't change the URL field in the service credentials. Doing so could lead to errors in your integration. Double check the URL [from our dedicated guide](#)³.

3.4.2 Step 3.2: Configure the service

1. Open Services tab
2. Open **worldline.https.direct.yourSiteId** and edit:
 - a. Name: replace **yourSiteId** with the actual ID for your site.
 - b. On test environments, the communication log could be enabled for debugging purposes.
 - c. Credentials: Check the updated **worldline.https.direct.yourSiteId.TEST** service credentials is selected.
3. Select **Apply** in the bottom-right corner of the screen.



To configure the service for LIVE credentials you need to:
 Follow step 3.1 for the **worldline.https.direct.yourSiteId.LIVE** credentials
 Follow step 3.2 selecting **worldline.https.direct.yourSiteId.LIVE** credentials in 3.2.2.c

3.5 Step 4: Set up Worldline-Direct Payments Cartridge

3.5.1 Step 4.1: Set up Worldline-Direct Payments Cartridge Configuration

To set up the Worldline-Direct Payments Cartridge Configuration in the Business Manager, go to **Merchant tools > Site Preferences > Custom Preferences > Worldline-Direct: General Config**

1. In the **Merchant ID** box, enter your Worldline-Direct merchant ID (PSPID).
2. Set your **Operation code** to one of:
 - a. **SALE** - direct capture (set by default)
 The amount has been ordered to be paid out in one go. Successful transactions will have StatusCode=9
 - b. **FINAL_AUTHORIZATION** - delayed capture
 The amount is only blocked on your customer's card. Successful transactions will have StatusCode=5 (this is used when you wish to capture a transaction only after shipping the article)

³ <https://support.direct.ingenico.com/documentation/api/endpoints>



When **FINAL_AUTHORIZATION** is selected

- make sure that you capture them later. Only then the transaction will reach StatusCode=9, for which you receive the payment for the transaction.
- make sure to enable and configure the [Automatic Capture Job](#) (see page 43) or to implement a **custom capture procedure** on top of this cartridge.
- you can manually capture funds fully / partially by opening an individual order once you have delivered the goods/services

3. Set **Checkout type** to one of:
 - a. Hosted Checkout Page (set by default) - read more information for [Hosted Checkout Page](#)⁴.
 - b. Hosted Tokenization Page - read more information for [Hosted Tokenization Page](#)⁵.
4. Configure 3DS settings:
 - a. Set **Enforce Strong Customer Authentication** to Yes or No.
 - b. Set **Exemption for transactions under 30 EUR** to Yes or No.



If both of the options are set to Yes, **Enforce Strong Customer Authentication** takes precedence and no transactions will be exempt.

5. Set **Apply Surcharge** to Yes or No.



Surcharge must be enabled on your Worldline account first or all payment requests will fail.

6. Set **Send line item prices to the API** to Yes or No.



It must be set to Yes for payment methods like Oney and Klarna.

7. Select **Save** in the upper-right corner of the screen.

3.5.2 Step 4.2: Configure Hosted Checkout Page Settings

To configure the Worldline-Direct Hosted Checkout Page settings in the Business Manager, go to **Merchant tools > Site Preferences > Custom Preferences > Worldline-Direct: Hosted Checkout Page Settings**

⁴ <https://support.direct.ingenico.com/documentation/integrate/hosted-checkout-page/>

⁵ <https://support.direct.ingenico.com/documentation/integrate/hosted-tokenization-page/>

1. If you have uploaded a [custom template](#)⁶ for Hosted Checkout Page in Worldline-Direct Backoffice, in the **Hosted Checkout Template** box, enter the name of the template file including .html suffix.
2. By default the cartridge lists all the credit card brands as separate payment methods. You can choose to group them together under single “Credit/Debit card” payment method by using the **“Groups Cards on the Hosted Checkout Page”** setting
3. Configure the **Hosted Checkout Session Timeout**. By default is set to 180 minutes.

3.5.3 Step 4.3: Configure Hosted Tokenization Page Settings

To configure the Worldline-Direct Hosted Checkout Page settings in the Business Manager, go to **Merchant tools > Site Preferences > Custom Preferences > Worldline-Direct: Hosted Tokenization Page Settings**

1. If you set your checkout type to Hosted Tokenization Page, in the **Hosted Tokenization JS** box enter the URL to the [Worldline-Direct Tokenizer Script](#)⁷.
2. If you have uploaded a [custom template](#)⁸ for Hosted Tokenization form in Worldline-Direct Backoffice, in the **Hosted Tokenization Template** box, enter the name of the template file including .html suffix.

3.5.4 Step 4.4: Configure Intersolve payment ID (optional)

The cartridge has the Intersolve payment method ID hardcoded to 5700. There is possibility that the Intersolve payment ID provided to you by Worldline is different. In such scenarios a small code change is needed.

1. Open the following file with an IDE: `int_worldline_direct/cartridge/scripts/worldline/direct/constants.js`
2. Locate the following line:

```
Constants.PAYMENT_PRODUCT_INTERSOLVE_ID = 5700;
```
3. Update the value with the provided Intersolve ID.

3.6 Step 5: Verify your connection to Worldline-Direct API

To test the connection with Worldline-Direct Direct API:

1. Go to **Merchant Tools > Ordering > Worldline-Direct Transactions**
2. Click on **Test API Connection** button in the upper-right corner of the screen.
3. If the set up is correct, an alert box **Connection to the Worldline-Direct API succeeded** should be displayed.

⁶ <https://support.direct.ingenico.com/documentation/template-builder/>

⁷ <https://support.direct.ingenico.com/documentation/integrate/hosted-tokenization-page/#validtokenisationURL>

⁸ <https://support.direct.ingenico.com/documentation/integrate/hosted-tokenization-page/#createiframetemplate>

4 Take payments with the cartridge

4.1 Payment methods

After the metadata import, two new payment methods have been added to the platform with their corresponding payment processors.

To make sure that this has been set up correctly:

1. In the Business Manager, go to **Merchant tools > Ordering > Payment Methods**.
2. In the list, check that **WORLDLINE_DIRECT_CARD**, **WORLDLINE_DIRECT_REDIRECT** and **WORLDLINE_DIRECT_CREDIT_REDIRECT** are created.

The payment methods added to the platform are placeholders only, no additional configuration is required and they are hidden from the end customer.

The list of customer's applicable payment methods is retrieved in real-time API request to Worldline-Direct API during checkout. The cartridge maps Worldline-Direct payment methods to the payment methods in the platform as follows:

- Card payment methods, except Bancontact, are mapped to **WORLDLINE_DIRECT_CARD**
- The iDeal payment method is mapped to **WORLDLINE_DIRECT_CREDIT_REDIRECT**
- All other payment methods are mapped to **WORLDLINE_DIRECT_REDIRECT**



When **checkout mode** is set to **Hosted Tokenization Page**, the **WORLDLINE_DIRECT_CARD** method is visible on the storefront. It is being used to group Worldline-Direct card payment methods into a single credit card option for the customer.

The list of all available Worldline-Direct [payment methods](#)⁹ can be found in Worldline-Direct Support Portal.

4.2 Taking payments

The Worldline-Direct Payments Cartridge ships pre-integrated to Worldline-Direct API.

It supports two of the most commonly used integration methods:

- Hosted Checkout Page
- Hosted Tokenization Page

⁹ <https://support.direct.ingenico.com/payment-methods/view-by-payment-product/>

4.2.1 Hosted Checkout Page

Hosted Checkout Page is the ideal solution to offer your customers online payments in your Salesforce Commerce Cloud store. The cartridge will redirect your customers from your checkout page to the Worldline-Direct platform for entering payment data.

Hosted Checkout Page supports all applicable Worldline-Direct payment methods.

Please read the [Hosted Checkout Page](#)¹⁰ documentation in Worldline-Direct Support Portal.

4.2.2 Hosted Tokenization Page

Worldline-Direct Hosted Tokenization Page solution offers you maximum security and flexibility at once. The cartridge will include an iFrame payment form in your checkout page. Worldline-Direct host the iFrame safely at their end, but you control its look and feel. The payment data entered by your customers is saved as a token, which you can use to make payments.

Hosted Checkout Page supports all Worldline-Direct card payment methods.

Please read the [Hosted Tokenization Page](#)¹¹ documentation in Worldline-Direct Support Portal.

4.2.3 Sending line item prices to the API

The Worldline API allows the line item prices to be sent when creating a Hosted Checkout request. In this case, the hosted page will also display a breakdown of the order prices, instead of showing only the order total.

This is mandatory for the Oney and Klarna payment methods and if the line item prices are not sent, Oney and Klarna will not be available in the payment methods list.

This behaviour is controlled through the following site preference: **Worldline-Direct: Shared Config > Send line item prices to the API**.



It is important to note that due to certain technical limitations sending the line item prices could lead to rounding errors if promotions are applied to the order. For example **ProductA** with **3** quantities and an order-level promotion can have a price of **10 EUR**. Therefore each individual quantity will have a price of **3.3333... EUR** which cannot be properly sent to the API.

In order to avoid an API error, this is handled in the following way:

On the Worldline website the products will be shown as:

ProductA x2 = **3.33** EUR

ProductA x1 = **3.34** EUR (any rounding errors get added to the price here)

The order total will be the same as shown on the SFCC website, but on the Worldline website the product with a rounding error will be split into two groups.

¹⁰ <https://support.direct.ingenico.com/documentation/integrate/hosted-checkout-page/>

¹¹ <https://support.direct.ingenico.com/documentation/integrate/hosted-tokenization-page/>

4.3 PCI Compliance

Hosted Checkout Page and Hosted Tokenization Page allow you to keep a low PCI profile as Worldline-Direct is responsible for managing sensitive data.

The **compliance level** for Hosted Checkout Page and Hosted Tokenization Page is **SAQ A (14)**.

5 Manage payments with the cartridge

The Worldline-Direct Payments Cartridge ships with a Business Manager module where you can see Worldline-Direct payment details as well as perform manual actions on transactions.

[Merchant Tools](#) > Ordering > Worldline-Direct Transactions

Worldline-Direct Transactions

[Test API connection](#)

This page allows you to search for orders which have Worldline-Direct transactions by order number. Also there is possibility to search Worldline transaction by id.

Search Order							By Order Number	By Transaction ID
Order Number: <input type="text"/> Find								
Number	Order Date	Created By	Registration Status	Customer	Customer Email	Order Total	Worldline Amount	Payment Status Category
00000514	4/04/22 12:03 pm	Customer	Registered	Test1 User1	testuser1@demandware.com	£174.79	£174.79	AUTHORIZED Details
00000509	4/04/22 11:07 am	Customer	Unregistered	Test1 User1	testuser1@demandware.com	£90.45	£90.45	AUTHORIZED Details
00000506	4/04/22 11:04 am	Customer	Unregistered	Test1 User1	testuser1@demandware.com	£90.45	£90.45	AUTHORIZED Details
00000505	4/04/22 11:03 am	Customer	Unregistered	Test1 User1	testuser1@demandware.com	£90.45	£90.45	AUTHORIZED Details
00000504	4/04/22 11:03 am	Customer	Unregistered	Test1 User1	testuser1@demandware.com	£90.45	£90.45	AUTHORIZED Details
00000503	4/04/22 10:53 am	Customer	Unregistered	Test1 User1	testuser1@demandware.com	£90.45	£90.45	AUTHORIZED Details
00000502	4/04/22 10:52 am	Customer	Unregistered	Test1 User1	testuser1@demandware.com	£90.45	£90.45	AUTHORIZED Details
00000501	4/04/22 10:50 am	Customer	Unregistered	Test1 User1	testuser1@demandware.com	£90.45	£90.45	AUTHORIZED Details
00000401	4/04/22 10:38 am	Customer	Registered	Test1 User1	testuser1@demandware.com	£101.35	£101.35	AUTHORIZED Details
00000303	4/04/22 9:39 am	Customer	Registered	Test1 User1	testuser1@demandware.com	£33.91	£33.91	AUTHORIZED Details

Showing 1 - 10 of 10 items.

Contact the support team at support.ecom@ingenico.com



1 Worldline-Direct Transactions Business Manager Module

To access the Worldline-Direct Transactions in the Business Manager, go to **Merchant Tools > Ordering > Worldline-Direct Transactions**.

The Worldline-Direct Transactions module provides the ability to search by Order number or by Worldline-Direct Transaction ID.

Transactions matching the search criteria are displayed in a table with information about the order, customer and Worldline-Direct transaction.

5.1 Open Details Dialog for a transaction in the Business Manager

To view more details for the Worldline-Direct transaction, you need to:

1. Open Worldline-Direct Transactions module.
2. Find the transaction
3. Click on **Details** link in the last column

The **Details Dialog** will be shown providing you more information to the transaction and the ability to perform manual actions (capture, refund, cancel) against the transaction.

5.2 Capture transactions in the Business Manager

Worldline-Direct Transactions module allows you to perform manual captures against a transaction. The page also shows a list of all captures to the transaction.

i Captures are available only when transaction is created in **FINAL_AUTHORIZATION** operation code.
Transactions created in **SALE** operation code are captured directly at the time of the authorisation.

Merchant Tools > Orders

Worldline

This page allows you

Search Order

Order Number:

Number	Order
00000514	4/04/2
00000509	4/04/2
00000506	4/04/2
00000505	4/04/2
00000504	4/04/2
00000503	4/04/2
00000502	4/04/2
00000501	4/04/2
00000401	4/04/2
00000303	4/04/2

Showing 1 - 10 of 10

Contact the support

Worldline-Direct Payment Details

Order No:	00000514	Order Total:	£174.79
Order Status:	NEW	Transaction Amount:	174.79 GBP
Worldline Payment ID:	3187760829	Captured Amount:	150.00 GBP
Payment Status Category:	AUTHORIZED	Refunded Amount:	0.00 GBP

ORIGINAL TRANSACTION DETAILS:

Transaction ID	Amount	Currency Code	Status
3187760829_0	174.79	GBP	PENDING_CAPTURE / 5

Captures

Refunds

Capture ID	Amount	Currency Code	Status
3187760829_1	150.00	GBP	CAPTURE_REQUESTED / 91

Add a new capture

Amount:

24.79

GBP

Capture

Cancel transaction

2 Capture view in Details dialog

5.2.1 View all captures for a transaction

To view the list of all captures for a transaction, in the Business Manager:

- 1. Open the **Details Dialog** for the transaction.
- 2. Click on **Captures** tab (selected by default)

i The capture transactions are retrieved in real-time using Worldline-Direct Direct API and you can see the latest status of the capture request.

5.2.2 Manual capture a transaction

To perform a manual capture, in the Business Manager:

1. Open the **Details Dialog** for the transaction
2. Click on **Captures** tab (selected by default)
3. At the bottom of the dialog window, find the **Add a new capture** form.
4. In the **Amount** box, enter the amount you wish to capture - it could be **full** or **partial amount** of the transaction.
5. Click on **Capture** button next to the amount field.

When the capture request is submitted, the Details Dialog will refresh its contents and will display the new capture transaction in the captures list.

As part of the capture request the [Handle Capture Business Process \(see page 52\)](#) will be triggered.

5.2.3 Automatic capture a transaction

The Worldline-Direct Payments Cartridge ships with built-in scheduled job to automatically capture transactions after configurable amount of time.

To configure the job please read [Automatic Capture Job \(see page 43\)](#) documentation.

5.3 Refund transactions in the Business Manager

Worldline-Direct Transactions module allows you to perform manual refunds against a transaction. The page also shows a list of all refunds to the transaction.

The screenshot shows the 'Worldline-Direct Payment Details' dialog box. It contains the following information:

- Order No:** 00000514
- Order Status:** NEW
- Worldline Payment ID:** 3187760829
- Payment Status Category:** AUTHORIZED
- Order Total:** £174.79
- Transaction Amount:** 174.79 GBP
- Captured Amount:** 150.00 GBP
- Refunded Amount:** 50.00 GBP

Below this, there is a section for 'ORIGINAL TRANSACTION DETAILS' with a table:

Transaction ID	Amount	Currency Code	Status
3187760829_0	174.79	GBP	PENDING_CAPTURE / 5

There is a 'Cancel transaction' button next to the first transaction row.

Below the table, there are two tabs: 'Captures' and 'Refunds'. The 'Refunds' tab is active, showing a table of refund transactions:

Refund ID	Amount	Currency Code	Status
3187760829_2	50.00	GBP	REFUND_REQUESTED / 81

At the bottom, there is a form to 'Add a new refund' with an 'Amount' field (set to 100.00), a 'Currency Code' dropdown (set to GBP), and a 'Refund' button.

3 Refunds view in Details dialog

5.3.1 View all refunds for a transaction

To view the list of all refunds for a transaction, in the Business Manager:

1. Open the **Details Dialog** for the transaction.
2. Click on **Refunds** tab



The refund transactions are retrieved in real-time using Worldline-Direct Direct API and you can see the latest status of the refund requests.

5.3.2 Refund a transaction

To refund a transaction, in the Business Manager:

1. Open the **Details Dialog** for the transaction
2. Click on **Refunds** tab
3. At the bottom of the dialog window, find the **Add a new refund** form.

4. In the **Amount** box, enter the amount you wish to refund - it could be **full** or **partial amount** of the transaction.
5. Click on **Refund** button next to the amount field.

When the refund request is submitted, the Details Dialog will refresh its contents and will display the new refund transaction in the refunds list.

5.4 Cancel transaction within the Business Manager

To cancel a transaction (authorisation reversal), in the Business Manager:

1. Open the **Details Dialog** for the transaction
2. Find the **Payment Status** section in the upper-right corner of the **Details Dialog**
3. Click on the **Cancel transaction** button.

6 Receive payment updates - Webhooks

Webhooks are messages sent from the Worldline-Direct platform to your Salesforce Commerce Cloud storefront. These messages inform you about the result of your transactions or about status changes that happen at a later point (i.e. captures, refunds).

More information about webhooks can be found in [Worldline-Direct Support Site](#)¹²

6.1 Configure Salesforce Commerce Cloud to receive payment updates

6.1.1 Step 1: Set up webhooks in Worldline-Direct Backoffice and Salesforce Commerce Cloud

To set up your Commerce Cloud store to receive updates from Worldline-Direct API, you need to set up a **Webhook URL**, **Webhook Key** and **Webhook Secret** in the Worldline-Direct Backoffice and copy it over to the Business Manager.



The **Webhook URL** to be configured in Worldline-Direct Backoffice by default is:
<https://{domain}/on/demandware.store/Sites-{yourSiteId}-Site/{locale}/WorldlineDirect-Webhooks>

Information on how to set up an Webhooks in Worldline-Direct Backoffice can be found [here](#)¹³.

To copy the **Webhook Key** and **Webhook Secret** over to the Business Manager:

1. Go to **Merchant Tools > Site Preferences > Custom Preferences > Worldline-Direct: Webhooks Config**
2. In the **Webhooks Key ID** field, enter the webhook key ID created in Worldline-Direct Backoffice.
3. In the **Webhooks Key Secret** field, enter the webhook key secret created in Worldline-Direct Backoffice.
4. Select **Save** in the upper-right corner of the screen.

6.1.2 Step 2: Set up the Process Webhooks Job

To set up the **Process Webhooks Job** in the Business Manager:

¹² <https://support.direct.ingenico.com/documentation/api/webhooks>

¹³ <https://support.direct.ingenico.com/documentation/api/webhooks#configurewebhooks>

1. Go to **Administration > Operations > Jobs**
2. Click on **WorldlineProcessWebhooks** in the list
3. Open **Job Steps** tab
4. Make sure the **scope** is set to your **site**
5. Configure the schedule for the job based on your needs.



For near real-time processing a value between 1 and 5 minutes is recommended based on your order volume. By default, the job is scheduled to run every 5 minutes.

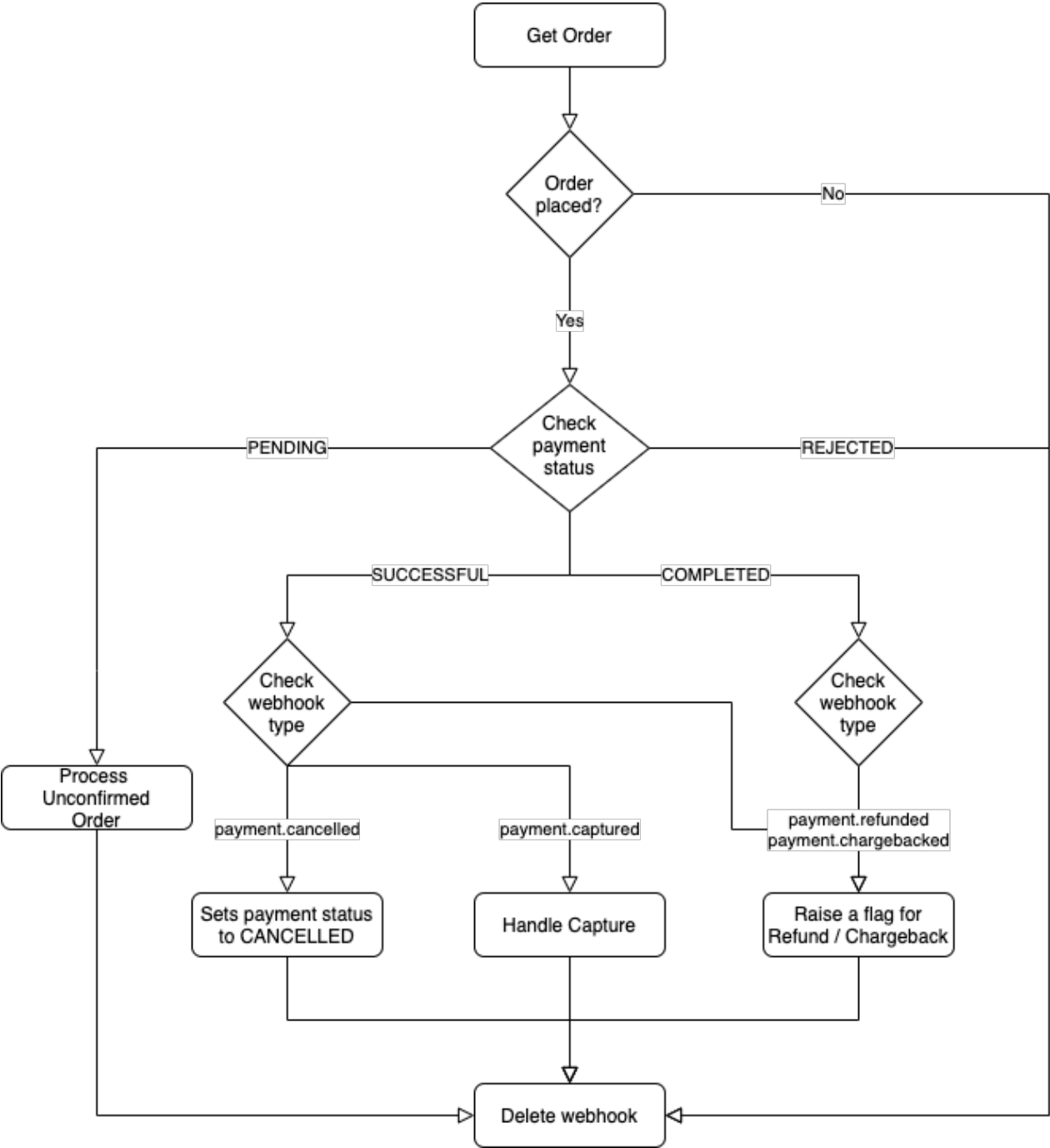
6.2 Webhook handling and processing

Whenever a message is received by the webhook endpoint, its signature is verified using the webhooks key and secret and if it's valid, then a new record is stored in WorldlineDirectWebhooks Custom Object.

The cartridge responds with HTTP 201 (Created) status code to notify Worldline-Direct that the webhook was received and it should not be sent again.

The webhook is later processed by the **Process Webhooks Job**.

6.2.1 Webhook event process flow



6.2.2 Business processes triggered by Process Webhooks Job

- [Process Unconfirmed Order Business Process \(see page 50\)](#)
- [Handle Capture Business Process \(see page 52\)](#)

6.2.3 Webhook events and process mapping

The table below describes the webhook events send to the system and actions performed for each of them.

Worldline-Direct API Event	Worldline-Direct API Event Description	Worldline-Direct Payment Cartiridge Action
payment.created	The transaction has been created. This is the initial state once a new payment is created	Not processed. Created payment is handled on customer redirection back to Commerce Cloud storefront after payment. Fallback: Payment Status Check Job.
payment.redirected	The consumer has been redirected to a 3rd party to complete the authentication/ payment	Not processed. Redirected payment is handled on customer redirection back to Commerce Cloud storefront after payment. Fallback: Payment Status Check Job.
payment.authorization_requested	We have requested an authorization against an asynchronous system and is awaiting its response (This is only applicable to Union Pay and Braspag.)	Not processed. Authorization requested payment is handled on customer redirection back to Commerce Cloud storefront after payment. Fallback: Payment Status Check Job.
payment.pending_approval	There are transactions waiting for your approval	Not processed. Pending approval payment is handled on customer redirection back to Commerce Cloud storefront after payment. Fallback: Payment Status Check Job.

payment.pending_completion	There are transactions waiting for you to complete them	Not processed. Pending completion payment is handled on customer redirection back to Commerce Cloud storefront after payment. Fallback: Payment Status Check Job.
payment.pending_capture	There are transactions waiting for you to capture them.	Not processed. Pending capture payment is handled on customer redirection back to Commerce Cloud storefront after payment. Fallback: Payment Status Check Job.
payment.capture_requested	The transaction is in the queue to be captured. (For Cards, this means that the transaction has been authorized.)	Not processed.
payment.captured	The transaction has been captured and we have received online confirmation	Processed. Triggers Handle Captures process.
payment.chargebacked	The transaction has been chargebacked	Processed. Raises flag "hasChargebacks" against the order.
payment.rejected	The transaction has been rejected	Not processed. Rejected payment is handled on customer redirection back to Commerce Cloud storefront after payment. Fallback: Payment Status Check Job.
payment.rejected_capture	We or one of our downstream acquirers/providers have rejected the capture request. For every 4xx and 5xx response an errorId and an array of errors is returned providing detailed error information.	Not processed.

payment.cancelled	You have cancelled the transaction	Processed. Sets the <code>worldlineDirectStatusCategory</code> custom order parameter to <code>CANCELLED</code> .
payment.refunded	The transaction has been refunded	Processed. Raises flag “hasRefunds” against the order.
refund.refund_requested	The transaction is in the queue to be refunded	Not processed.

6.2.4 Logging and debugging

The webhooks use the following custom log files:

Log file prefix	Log category	Description
worldline-webhooks	worldline-webhooks	Used by the controller that receives the webhooks. Set the “Log level” to DEBUG if you want to log each webhook that arrives. Otherwise set to WARN or ERROR (recommended for Production instances).
worldline-job-webhooks	worldline-jobs	Used by the scheduled job that processes the webhooks. Set the “Log level” to DEBUG if you want to log each webhook that gets processed. Otherwise set to WARN or ERROR (recommended for Production instances).

7 Test the cartridge

TBD: QA to provide test cases for storefront and Business Manager.

7.1 Test cases (Registered + saved cards/Guest user HCP)

Test Scenario	Outcome
1. Complete checkout with Amex (374111111111111). Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
2. Complete checkout with Amex (371449635311004). Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
3. Complete checkout with Dinners club (36255695580017). Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
4. Complete checkout with MasterCard 5137009801943438 3DS v2 Frictionless flow. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
5. Complete checkout with MasterCard 5130257474533310 3DS v2 Challenge flow. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)

Test Scenario	Outcome
6. Complete checkout with JCB 3528798062014879. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
7. Complete checkout with Maestro 50339619890917. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
8. Complete checkout with Maestro 50339619890917. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
9. Complete checkout with Visa 4330264936344675 3ds v2 Frictionless flow. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
10. Complete checkout with BCMC 670300000000000003. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
11. Complete checkout with iDEAL. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)

Test Scenario	Outcome
12. Complete checkout with ILLICADO. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
13. Complete checkout with OneyBrandedGiftCard. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)
13. Complete checkout with PayPal. Check the payment methods, HCP and check the order in BM	All payment methods, including cards, are listed as separate payment methods on the payment step. The user is redirected to HCP to finish the checkout. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)

7.2 Test cases (Registered + saved cards/Guest user HTP)

Test Scenario	Outcome
1. Complete checkout with AMEX (375418081197346) 3-D Secure V2 Frictionless flow. Check the payment methods, HTP and check the order in BM	All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)

Test Scenario	Outcome
<p>2. Complete checkout with Amex (379764422997381) 3-D Secure V2 Challenge flow Order. Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>
<p>3. Complete checkout with Dinners club (36255695580017). Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is not placed successfully, card number shown as incorrect or incompatible.</p>
<p>4. Complete checkout with MasterCard 5168693992589936) 3DS v2 Challenge flow Not enrolled/proof of attempt (ECI=6). Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>

Test Scenario	Outcome
<p>5. Complete checkout with JCB 3528798062014879 3DS v2 Challenge flow. Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>
<p>6. Complete checkout with Maestro 50339619890917. Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>
<p>7. Complete checkout with Visa 4874970686672022 3DS v2 Challenge flow. Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>

Test Scenario	Outcome
<p>8. Complete checkout with BCMC 67030000000000003. Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>
<p>10. Complete checkout with BCMC 67030000000000003. Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after providing the card data in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>
<p>11. Complete checkout with iDEAL. Check the payment methods, HTP and check the order in BM</p>	<p>All card types are bundled together for selection as a single payment method. Bancontact QR code is displayed as a separate payment method. The remaining non-card payment methods are listed as separate payment methods. The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after selecting the payment method in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>

Test Scenario	Outcome
12. Complete checkout with ILLICADO. Check the payment methods, HTP and check the order in BM	<p>All card types are bundled together for selection as a single payment method.</p> <p>Bancontact QR code is displayed as a separate payment method.</p> <p>The remaining non-card payment methods are listed as separate payment methods.</p> <p>The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after selecting the payment method in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>
13. Complete checkout with OneyBrandedGiftCard. Check the payment methods, HTP and check the order in BM	<p>All card types are bundled together for selection as a single payment method.</p> <p>Bancontact QR code is displayed as a separate payment method.</p> <p>The remaining non-card payment methods are listed as separate payment methods.</p> <p>The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after selecting the payment method in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>
13. Complete checkout with PayPal. Check the payment methods, HTP and check the order in BM	<p>All card types are bundled together for selection as a single payment method.</p> <p>Bancontact QR code is displayed as a separate payment method.</p> <p>The remaining non-card payment methods are listed as separate payment methods.</p> <p>The additional parameters for non-card payment methods (if applicable) are also displayed for a customer to make a selection. The user can proceed to finish the checkout, after selecting the payment method in the iFrame on the payment step. The order is placed successfully with all relevant data reflected for the order in BM (transaction status, payment method, etc.)</p>

8 Appendix 1: Salesforce Commerce Cloud data model extensions

8.1 System Objects

8.1.1 Site Preferences

Attribute ID	Attribute Group	Description
worldlineDirectMerchantID	WORLDLINE_DIRECT	Merchant ID for Worldline-Direct platform
worldlineDirectOperationCode	WORLDLINE_DIRECT	The operation code will be valid for all transactions: <ul style="list-style-type: none"> • SALE: capture the payments when the order is placed • FINAL_AUTHORIZATION: capture later (manually or through a job)
worldlineDirectCheckoutSendLineItemPrices	WORLDLINE_DIRECT	Whether line item prices will be sent to the API or not. For details, please see Sending line item prices to the API (see page 0)
worldlineDirectCheckoutType	WORLDLINE_DIRECT_CREDIT	Checkout type for card payments: Hosted Checkout Page (Redirect) OR Hosted Tokenization Page (iFrame)
worldlineDirectHTJS	WORLDLINE_DIRECT_CREDIT	Please provide the URL to Hosted Tokenization JS Library

Attribute ID	Attribute Group	Description
worldlineDirectHTPTemplate	WORLDLINE_DIRECT_CREDIT	Please provide the template name as in Worldline-Direct Backoffice to be rendered to the customer
worldlineDirectHCPTemplate	WORLDLINE_DIRECT_REDIRECT	Please provide the template name as in Worldline-Direct Backoffice to be rendered to the customer
worldlineDirectWebhooksKeyID	WORLDLINE_DIRECT_WEBHOOKS	The webhooks key id that should be copied from Worldline-Direct's backoffice
worldlineDirectWebhooksKeySecret	WORLDLINE_DIRECT_WEBHOOKS	The webhooks key secret that should be copied from Worldline-Direct's backoffice

8.1.2 Order

Attribute ID	Attribute Group	Description
isWorldlineDirectOrder	WORLDLINE_DIRECT	A flag to mark orders that have an Worldline-Direct payment
worldlineDirectTransactionID	WORLDLINE_DIRECT	The Worldline-Direct transaction ID
worldlineDirectStatusCategory	WORLDLINE_DIRECT	Payment status category
worldlineDirectStatus	WORLDLINE_DIRECT	Payment status
worldlineDirectStatusCode	WORLDLINE_DIRECT	Transaction status code
worldlineDirectIsAuthorized	WORLDLINE_DIRECT	Is the transaction authorized
worldlineDirectIsCancellable	WORLDLINE_DIRECT	Is the transaction cancellable

Attribute ID	Attribute Group	Description
worldlineDirectIsRefundable	WORLDLINE_DIRECT	Is the transaction refundable
worldlineDirectHasCaptures	WORLDLINE_DIRECT	Whether there have been captures for this order
worldlineDirectHasRefunds	WORLDLINE_DIRECT	Whether there have been refunds for this order
worldlineDirectHasChargebacks	WORLDLINE_DIRECT	Whether there have been chargebacks for this order

8.1.3 Customer Payment Instrument

Attribute ID	Attribute Group	Description
worldlineDirectPaymentProductID	WORLDLINE_DIRECT	Worldline-Direct Payment Product ID
worldlineDirectPaymentProductName	WORLDLINE_DIRECT	Worldline-Direct Payment Product Name
worldlineDirectPaymentMethod	WORLDLINE_DIRECT	Worldline-Direct Payment Method
worldlineDirectCreditCardAlias	WORLDLINE_DIRECT	The masked credit card number given by the Worldline API

8.1.4 Order Payment Instrument

Attribute ID	Attribute Group	Description
worldlineDirectPaymentProductID	WORLDLINE_DIRECT	Worldline-Direct Payment Product ID

Attribute ID	Attribute Group	Description
worldlineDirectPaymentProductName	WORLDLINE_DIRECT	Worldline-Direct Payment Product Name
worldlineDirectPaymentMethod	WORLDLINE_DIRECT	Worldline-Direct Payment Method
worldlineDirectCreditCardAlias	WORLDLINE_DIRECT	The masked credit card number given by the Worldline API
worldlineDirectPaymentDirectoryIssuerID	WORLDLINE_DIRECT	Worldline-Direct Issuer ID
worldlineDirectCardExpiry	WORLDLINE_DIRECT	Card Expiry
worldlineDirectAuthorisationCode	WORLDLINE_DIRECT	Authorisation Code
worldlineDirectHostedCheckoutID	WORLDLINE_DIRECT	Hosted Checkout ID

8.2 Custom Objects

8.2.1 WorldlineDirectWebhooks

When a [webhook](#)¹⁴ arrives, first its signature will be checked and if it's valid, it will get stored in this custom object. It will later be processed by the [Process Webhooks Events Job](#)¹⁵. Once processed, the custom object will be deleted.

Attribute ID	Type	Description
body	String	Stores the body of the webhook.

¹⁴ <https://support.direct.ingenico.com/documentation/api/webhooks>

¹⁵ <https://greenlight-digital.atlassian.net/wiki/pages/createpage.action?fromPageId=3169583199&linkCreation=true&spaceKey=ING&title=Process+Webhook+Events+Job>

8.2.2 WorldlineDirectPaymentProducts

The plugin allows the merchant to hide certain Worldline payment methods from the billing page. In order to achieve this, the list of payment methods are stored in a custom object and each one has a boolean property that specifies if it should be hidden from the billing page or not.

Attribute ID	Type	Description
entriesJSON	Text	Stores the list of Worldline payment products in JSON format.

9 Appendix 2: Scheduled Jobs

Job ID	Description	Default schedule
WorldlineCaptureAuthorizedPayments	<p>Captures the payments that are awaiting capture. It can be configured to either capture all awaiting payments or capture only the payments that have been made more than X days ago (using the worldlineDirectCaptureProcedure Delay parameter).</p> <p>More information can be found here: Automatic Capture Job (see page 43)</p>	Every day at 11:00 pm server time
WorldlinePaymentStatusCheck	<p>Checks the Worldline-Direct status for payments in Pending status or orders in status Created and updates the order.</p> <p>More information can be found here: Check Payment Status Job (see page 42)</p>	Every hour
WorldlineProcessWebhooks	<p>Takes the webhooks that were previously stored in the WorldlineDirectWebhooks custom object and processes them.</p> <p>More information can be found here: Receive payment updates - Webhooks (see page 23)</p>	Every 5 minutes

9.1 Check Payment Status Job

9.1.1 Description

This job consists of two parts:

- processes orders that are in status “CREATED” (which means “incomplete”), but have a successful payment
- processes orders whose Worldline-Direct payment status is still not finalised and is waiting for a certain update (unconfirmed orders)

Detailed explanation of each part is given below.

9.1.1.1 Processing orders in status “CREATED” which have a successful payment

This can happen in case the user started checkout, got redirected to the Worldline-Direct site, completed the payment there and got charged, but for some reason, such as a network error, did not get redirected back to the SCC website to complete the order (i.e. the user did not see the “thank you for your order” page, which also completes the order on SCC side).

The scheduled job will try to finalise these orders. To do that, it will look for orders that are in status “**CREATED**”, have a successful payment and have been created more than 1 hour ago. The 1 hour period is added to make sure the job does not mistakenly pick up orders that are currently being placed.

The job will then validate the payment and if it was successful, it will finalise the order, so that it will become in status “**NEW**” or “**OPEN**”.

9.1.1.2 Processing unconfirmed orders

The job will pick up all unconfirmed orders (order status is not in “**CREATED**” or “**FAILED**” and confirmation status is “**Not Confirmed**”). For each order the job will make a request to check the latest payment status:

- If the payment status is successful, the job will update the confirmation status to “**Confirmed**”.
- If the payment status is rejected, the job will cancel the order.
- If the payment status is still pending and the order is older than 30 hours (the number of hours is configurable with a default value of 30), the job will cancel the order and the payment transaction.

A detail description of the process can be found in [Process Unconfirmed Order Business Process \(see page 50\)](#).

9.1.2 Configuration

The job uses the following configuration parameters:

Parameter Name	Default Value	Description
cancelUnconfirmedOrderAfterHours	30	How many hours to wait before cancelling an order with a pending Worldline-Direct payment status.

9.1.3 Schedule

The job is preconfigured to run once per hour in order to regularly receive payment updates. It is up to the merchant to determine if this default needs to be changed in order to fit their business requirements.

9.2 Automatic Capture Job

9.2.1 Description

This scheduled job can perform automatic captures of payments that have not been captured yet. This is typically needed when the payments do not get captured automatically when placing the order (i.e. when the “Operation code” site preference is set to “FINAL_AUTHORIZATION”) and a mechanism is needed to perform these captures at a later stage.

There are cases in which the merchant may decide that this job is not needed and can be disabled:

- If the payment gets automatically captured as soon as the order is placed (i.e. when the “Operation code” site preference is set to “SALE”).
- If the merchant wants to perform the captures manually

Once a payment gets picked up by this job, the following process gets executed:

[Handle Capture Business Process \(see page 52\)](#)

9.2.2 Configuration

The job uses the following parameters:

Parameter name	Description	Configuration steps
worldlineDirectCaptureProcedureDelay	How many days to wait before capturing a pending payment	Administration > Jobs > WorldlineCaptureAuthorizedPayments > Job Steps > WorldlineCaptureAuthorizedPayments

How frequently the job should run and whether it should wait for a specific number of days before capturing a payment depends on the specific business case and is up to the merchant to decide.

Two typical use cases are described below as well as the configurations needed to achieve them.

9.2.2.1 Capture payments, placed more than X days ago

In this case the merchant may decide to run this job several times per day. This can be achieved with the following steps:

1. Go to Business Manager, then **Administration > Jobs > WorldlineCaptureAuthorizedPayments > Schedule and History**
2. Make sure the “Enabled” checkbox is checked
3. Set “Trigger” to “Recurring Interval”
4. The job can be scheduled to run every N hours by setting “Interval” to “Hours” and “Amount” to N
5. The initial hour can be specified in the “From” field. For example: if configured to run every 3 hours and the “From” field specifies “01:00:00” as an hour, then the job will be executed at 01:00:00, 04:00:00, 07:00:00, etc. These hours will be in the server’s timezone which can be seen at the bottom of the Business Manager page (“Instance Time Zone: ...”).

Additionally the `worldlineDirectCaptureProcedureDelay` parameter needs to be set to the correct value:

- If it is set to 1, then all pending payments made more than 1 day ago will be captured
- If it is set to 2, then all pending payments made more than 2 days ago will be captured
- ...

9.2.2.2 Capture all pending payments once per day, at the end of the day

This can be achieved with the following steps:

1. Go to Business Manager, then **Administration > Jobs > WorldlineCaptureAuthorizedPayments > Schedule and History**
2. Make sure the “Enabled” checkbox is checked
3. Set “Trigger” to “Recurring Interval”
4. Set the “Amount” field to “1” and “Interval” to “Days”
5. The exact hour can be specified in the “From” field. This hour will be in the server’s timezone which can be seen at the bottom of the Business Manager page (“Instance Time Zone: ...”).

Additionally the `worldlineDirectCaptureProcedureDelay` parameter can be set to 0, in this case all pending payments will be captured.

9.2.3 Schedule

The job ships preconfigured to run once per day at 23:00 server time (the timezone can be seen at the bottom of the page in SCC Business Manager).

By default the `worldlineDirectCaptureProcedureDelay` parameter is set to 1, which means that payments made more than 1 day ago will be captured.

It is up to the merchant to determine if these defaults need to be changed to fit their specific business requirements.

10 Appendix 3: Business processes

10.1 Validate Payment Business Process

10.1.1 Description

Worldline-Direct Payments Cartridge validates the payment response after customer has made a payment attempt.

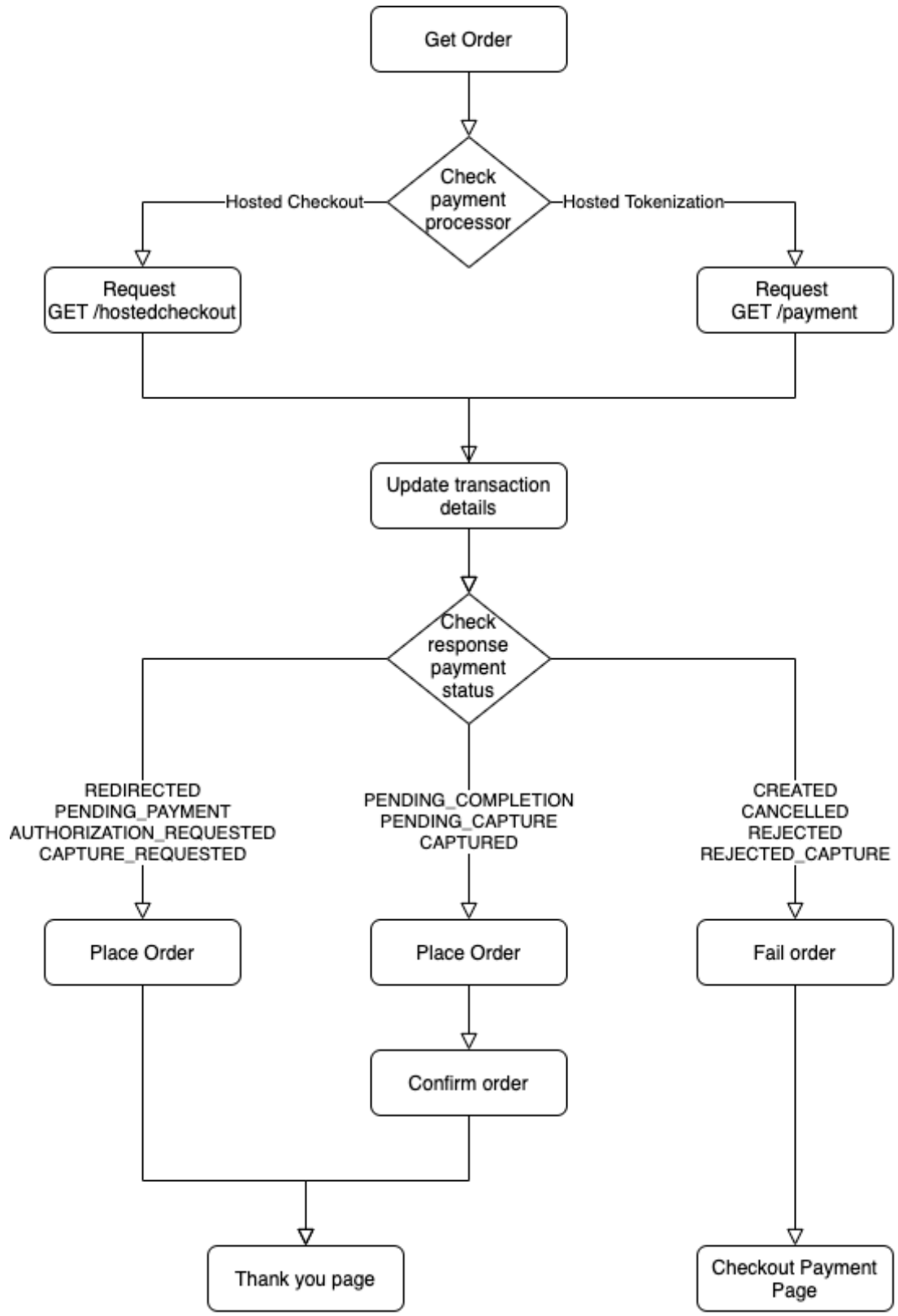
The business process is triggered by following actions:

- Customer is redirected back to Commerce Cloud storefront from Hosted Checkout Page
- Customer is redirected back to Commerce Cloud storefront from 3DS verification
- Customer has completed a payment on Hosted Tokenization flow without redirect

10.1.2 Steps

1. Get customer's order
2. Get payment instrument
3. Make a request to Worldline-Direct to retrieve the actual payment status
4. Save payment details against the payment instrument
5. Check payment status
 - a. When status is one of REJECTED statuses the process **fails the order**, restores customer baskets and redirects the customer to the Checkout: Payment page with an user-friendly error message
 - b. When status is one of SUCCESSFUL statuses the process **places the order**, sets Confirmation Status to **Confirmed** and redirects the customer to the Thank you page.
 - c. When status is one of STATUS_UNKNOWN statuses the process **places the order**, sets Confirmation Status to **Not Confirmed** and redirects the customer to the Thank you page.

10.1.3
Flowchart



10.1.4 Worldline-Direct Payment statuses mapping

The table below showcase the mapping between Worldline-Direct API Payment Statuses, Worldline-Direct Payment Cartridge statuses and Salesforce Commerce Cloud Order statuses used in Validate Payment Business Process.

Worldline-Direct API Status Category	Worldline-Direct API Status	Worldline-Direct Payment Cartridge Status Category	Order Status	Order Confirmation Status	Storefront Customer landing page
REJECTED	CREATED	REJECTED	FAILED	Not Confirmed	Checkout: Payment page with error message
	CANCELLED				
	REJECTED				
	REJECTED_CAPTURE				
STATUS_UNKOWN	REDIRECTED	STATUS_UNKOWN	NEW		
SUCCESSFUL	AUTHORIZATION_REQUESTED				
	CAPTURE_REQUESTED				
	PENDING_PAYMENT				
	PENDING_COMPLETION	SUCCESSFUL		Confirmed	
	PENDING_CAPTURE				
	CAPTURED				

10.2 Process Unconfirmed Order Business Process

10.2.1 Description

Worldline-Direct Payments Cartridge implements business process to process the unconfirmed orders in Salesforce Commerce Cloud platform.

The business process is triggered by following actions:

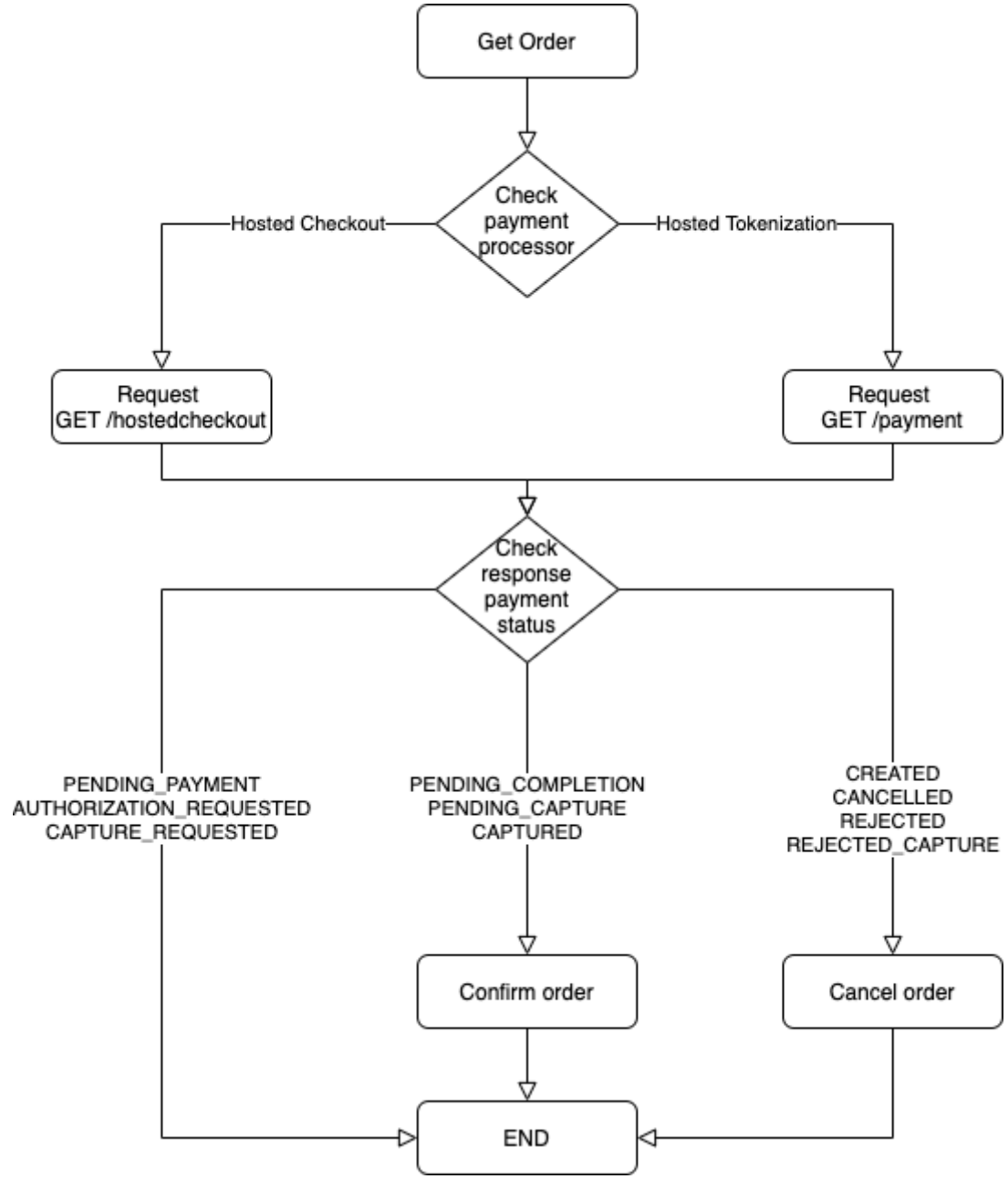
- [Check Payment Status Job](#) (see page 42) is processing an unconfirmed order
- [Process Webhook Events Job](#)¹⁶ is processing an unconfirmed order

10.2.2 Steps

1. Get the order
2. Get the payment transaction
3. Make a request to Worldline-Direct API to retrieve the latest available payment status
4. Save payment details against the payment instrument
5. Check payment status
 - a. When status is one of REJECTED statuses the process **cancels the order**.
 - b. When status is one of SUCCESSFUL statuses the process sets Confirmation Status to **Confirmed**.
 - c. When status is one of STATUS_UNKNOWN statuses the process **does nothing**.

¹⁶ <https://greenlight-digital.atlassian.net/wiki/pages/createpage.action?fromPageId=3159720503&linkCreation=true&spaceKey=ING&title=Process+Webhook+Events+Job>

10.2.3
Flowchart



10.2.4 Worldline-Direct API Payment statuses mapping

The table below showcase the mapping between Worldline-Direct API Payment Statuses, Worldline-Direct Payment Cartridge statuses and Salesforce Commerce Cloud Order statuses used in Process Unconfirmed Order Business Process.

Worldline-Direct API Status Category	Worldline-Direct API Status	Worldline-Direct Payment Cartridge Status Category	Order Status	Order Confirmation Status
REJECTED	CREATED	REJECTED	Cancelled	Not Confirmed
	CANCELLED			
	REJECTED			
	REJECTED_CAPTURE			
STATUS_UNKNOWN	REDIRECTED	STATUS_UNKNOWN	Not modified	
SUCCESSFUL	AUTHORIZATION_REQUESTED			
	CAPTURE_REQUESTED			
	PENDING_PAYMENT			
	PENDING_COMPLETION	SUCCESSFUL		
	PENDING_CAPTURE			
CAPTURED				

10.3 Handle Capture Business Process

10.3.1 Description

Worldline-Direct Payments Cartridge implements business process to monitor the successful captures against an order.

The business process is triggered by following actions:

- Merchant is requesting [manual capture](#)¹⁷ using Business Manager

¹⁷ <https://greenlight-digital.atlassian.net/wiki/pages/createpage.action?fromPageId=3159720475&linkCreation=true&spaceKey=ING&title=Captures>

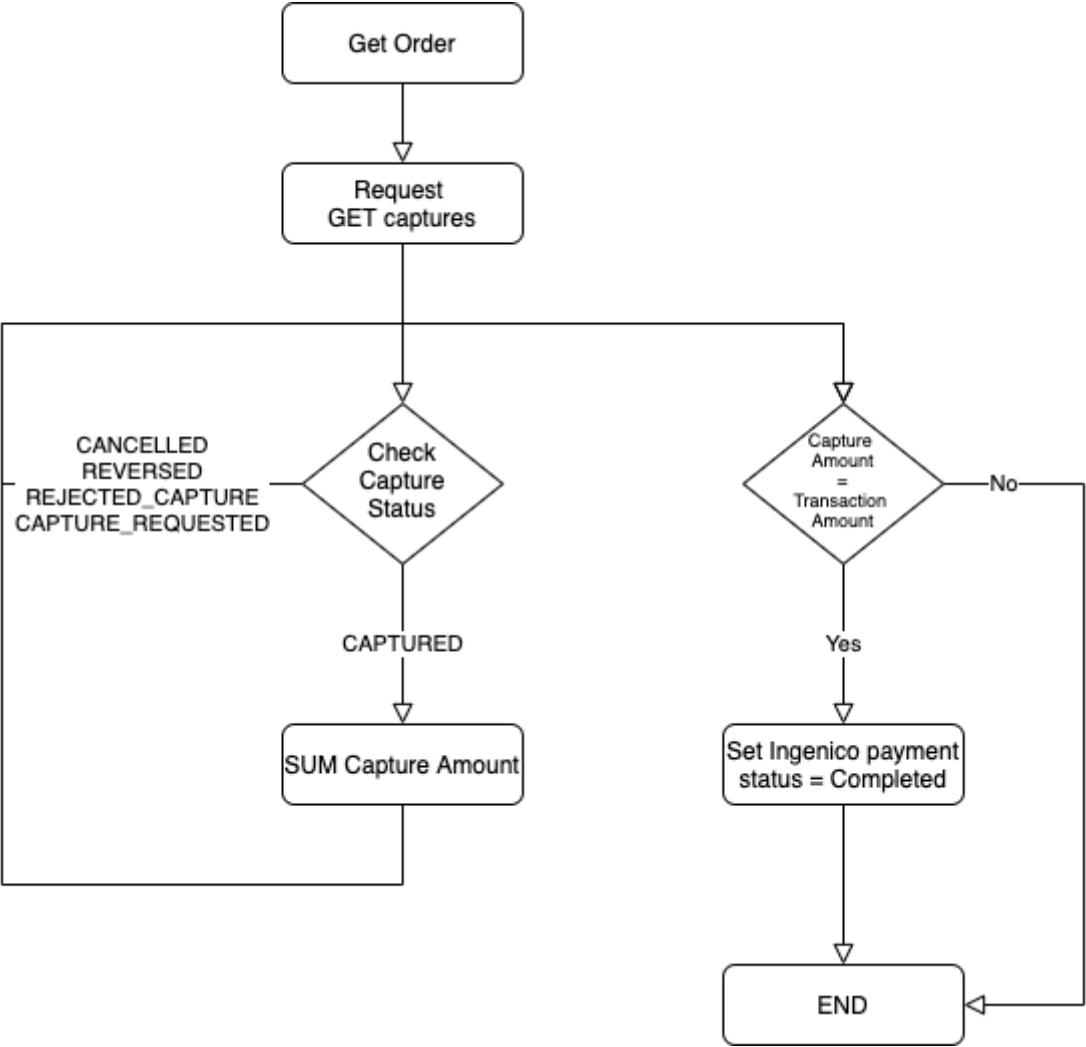
- Cartridge is requesting automatic capture via [Automatic Capture Job](#) (see page 43)
- [Process Webhook Events Job](#)¹⁸ is processing a “payment.captured” event

10.3.2 Steps

1. Get the order
2. Get the payment transaction
3. Make a request to Worldline-Direct API to retrieve all captures for this payment
4. Parse the response and for each “CAPTURED” capture sums the captured amount
5. When captured amount equals the transaction amount sets the order’s “Worldline Payment Status Category” to COMPLETED.

¹⁸ <https://greenlight-digital.atlassian.net/wiki/pages/createpage.action?fromPageId=3159720475&linkCreation=true&spaceKey=ING&title=Process+Webhook+Events+Job>

10.3.3
Flowchart



10.3.4 Worldline-Direct API Capture statuses mapping

The table below showcase the mapping between Worldline-Direct API Payment Statuses, Worldline-Direct Payment Cartridge statuses and Salesforce Commerce Cloud Order statuses used in Handle Capture Business Process.

Worldline-Direct API Status Category	Worldline-Direct API Status	Worldline-Direct Payment Cartridge Status Category
UNSUCCESSFUL	CANCELLED	Not modified
	REJECTED_CAPTURE	
	REVERSED	
PENDING_CONNECT_OR_3RD_PARTY	CAPTURE_REQUESTED	
COMPLETED	CAPTURED	COMPLETED

11 Appendix 4: Services

Service Name	Type	Description
worldline.https.direct.yourSiteID	HTTP	The main service used to communicate to Worldline-Direct API.

12 Appendix 5: Failover and Recovery

If during the checkout process communication between Salesforce and Worldline fails for some reason (server outage, misconfiguration, etc.) an error will be displayed on the checkout page.

Additionally all communication errors are recorded in the server logs where the merchant is able to examine the exact error.

12.1 Log files

The main communication between Salesforce and the Worldline API is logged in the service log that is configured in **Administration > Operations > Services > worldline.https.direct.yourSiteId**

If the checkbox **“Communication Log Enabled”** is enabled, then the API requests and responses will be available in the log file `“service-worldline- ...”`.

Additional log settings can be configured from **Administration > Operations > Custom Log Settings**

The cartridge uses two additional log categories, besides the standard root category: **“worldline-jobs”** and **“worldline-webhooks”** for the jobs and webhooks respectively.

Log levels can be configured for each category.

13 Appendix 6: Upgrading the cartridge

Currently two major versions of this cartridge exist: **21.*** and **23.***. If an upgrade from version **21.*** to **23.*** is needed, this document provides a checklist with needed steps.

Version **23.*** introduces many code changes, including a rebranding and as a result the easiest way to upgrade from version **21.*** is to delete the previous cartridge and reimport the new one.

13.1 Deleting version 21.* code

The following cartridges need to be deleted:

- **int_ingenico_ogone**
- **bm_ingenico_ogone**

13.2 Deleting version 21.* metadata

A checklist of metadata that needs to be deleted can be found below:

Administration > Operations > Services: All services, profiles and credentials that start with “**ingenico**”

Administration > Operations > Jobs: All jobs that start with “**Ingenico**”

Administration > Site Development > Custom Object Types: All types that start with “**Ingenico**”

Administration > Site Development > System Object Types > CustomerPaymentInstrument: All attributes and groups that start with “**ingenico**”

Administration > Site Development > System Object Types > OrderPaymentInstrument: All attributes and groups that start with “**ingenico**”

Administration > Site Development > System Object Types > Order: All attributes and groups that start with “**ingenico**”

Administration > Site Development > System Object Types > SitePreferences: All attributes and groups that start with “**ingenico**”

Merchant Tools > [yourSiteId] > Payment Methods: All payment methods that start with “**INGENICO**”

Merchant Tools > [yourSiteId] > Payment Processors: All payment processors that start with “**INGENICO**”

13.3 Importing the new version

Once the previous version has been deleted, the new one can be imported as described here: [Install the cartridge \(see page 7\)](#)