



**PREPARED BY**

Tamara

Salesforce B2C Developer Certification



# EXECUTIVE SUMMARY

[Tamara is a Buy Now Pay Later solution](#) for brick and mortar and e-commerce players in the GCC.

Our solution integrates with the merchant's checkout. It tackles cash on delivery challenges and increases order frequency, average order size and conversion. It also provides consumers with an exceptional checkout experience.

Document version	Date	Description
1.0.0	2021/01/13	Initial version
1.0.1	2021/04/21	2nd version
1.0.2	2021/06/03	Remove unnecessary attributes in Payment Transaction plus added configuration for Tamara widget
1.0.3	2021/09/13	Support Pay in 6 and Tamara orders management page

## 1. Project Overview

- The project objective is to build a Salesforce cartridge to integrate TAMARA payment solution with an SFCC site. Thus, the cartridge can promote the TAMARA payment solution application and help merchants & developers to easily integrate TAMARA payment into their SFCC e-commerce sites.
- This cartridge will be built based on the latest Salesforce template using [SFRA 5.1.0](#) and JSController.
- The user is redirected to the Tamara hosted page to collect the payment details.
- Authorization during the checkout process and only capture the payment after the product shipped.
- The return/cancel process is not included as this is not the standard flow in the commerce cloud. However, the cartridge provides the interface to call the Cancel/Refund API to Tamara via Service Framework. Partners flexible to leverage them into your application.

## 2. Limitations And Constraints

This section details the limitations, constraints and best practice for merchants planning to integrate the Tamara cartridge with SFCC Storefront.

- The cartridge doesn't support the cancellation or refund process following the SFCC standard. However, there are the Service interfaces that are developed based on the Service Framework in SFCC for Cancel/Refund. A merchant needs to leverage them into their application in order to archive these flows.
- Merchant shall take care of Site Custom pretence and services credentials settings. *(The detail is mentioned in the below section).*

## 3. Cartridges Description

.No	Cartridge	Description
1	int_tamara_core	The global definition of the Tamara cartridge such as Global constant, Global Job, Global Text, Global template, etc...

2	int_tamara_sfra	The customization on top of the SFRA template.
3	bm_tamara	Provide cancelation/refund (full/partial) from BM

## 4. Implementation Guide

This section describes the detailed step to install the Tamara cartridge into your current site which is based on the SFRA template.

### 4.1 Installation

#### 4.1.1 Import metadata

Upload and import metadata.zip.

To do so, go to Business Manager > Administration > Site Development > Site Import & Export

Upload archive using Local option in the Upload Archive section. After upload select the metadata/metadata.zip file in the list and click on the import button.

Administration > Site Development > Site Import & Export ← 1. Go to Site Import

#### Site Import & Export

This page allows you to export the current configuration of your organization including all of its sites. To download an archive, just click its file name.

### Import

Upload Archive:

☒ Local ☐ Remote

Choose File No file chosen Upload

3. Select metadata/metadata.zip file in the repository And upload

4. Import

Select	Name ▲	Location	File Size	Last Modified
<input type="radio"/>	SiteGenesis Demo Site			
<input type="radio"/>	Storefront Reference Architecture Demo Sites			


Import Delete

#### 4.1.2 Build code

- Go to the root of the SFRA source code
- Run ***npm install***

- Go to package.json > look at “paths.base” property > change the “base” to the relative path of SFRA source folder.

```
{
  "name": "int_tamara_sfra",
  "version": "0.0.1",
  "description": "Tamara payment",
  "main": "index.js",
  "paths": {
    "base": "../storefront-reference-architecture/cartridges/app_storefront_base/"
  },
}
```



- Run **npm run build**
- Upload those cartridges int\_tamara\_core and int\_tamara\_sfra bm\_tamara to the SFCC environment

### 4.1.3 Cartridge Path

Add int\_tamara\_core and int\_tamara\_sfra into the cartridge path of a target site. Ensure to put this int\_tamara\_sfra:int\_tamara\_core in front of the current cartridge path.

To do so, go to **Business Manager > Administration > Sites > Manage Sites > Your Target Site > 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 ("SEO > Aliases Configuration"). The HTTP/HTTPS hostname hostnames are defined by aliases configuration and are intended only to support an older configuration style.

HTTP Hostname:


HTTPS Hostname:

Instance Type: All

Cartridges: int\_tamara\_sfra:int\_tamara\_core:plugin\_cartridge\_merge:plugin\_instorepickup:plu

Effective Cartridge Path:

- plugin\_cartridge\_merge
- plugin\_instorepickup
- plugin\_wishlists
- plugin\_giftregistry
- lib\_productlist
- plugin\_productcompare
- plugin\_sitemap
- plugin\_applepay
- plugin\_datadownload
- app\_storefront\_base
- plugin\_apple\_pay
- plugin\_facebook
- plugin\_payments
- plugin\_pinterest commerce



And then go to **Administration > Sites > Manage Sites > Business Manager - Settings**. Do the similar step, add the **bm\_tamara:int\_tamara\_core:app\_storefront\_base:bm\_app\_storefront\_base:modules** in front of the current cartridge path.

#### 4.1.4 Service Endpoints

Sandbox	<a href="https://api-sandbox.tamara.co">https://api-sandbox.tamara.co</a>
Production	Contact Tamara to get it ( <a href="#">more detail in section 4.2.2</a> )

## 4.2 Configuration

### 4.2.1 Custom Preference

Updating Cartridge Custom Site Preferences

Go to **Business Manager > Merchant Tools > Site Preferences > Custom Site Preferences > Tamara Payment**.

Merchant Tools / Site Preferences / **Custom Site Preference Groups** [View Across Sites](#)

Search by ID

ID	Name	Description	Preferences	View Across Sites
<a href="#">Storefront Configs</a>	Storefront Configurations		5	<a href="#">View</a>
<a href="#">SFRA Unified Feature Cartri...</a>	SFRA Unified Feature Cartridge		6	<a href="#">View</a>
<a href="#">Tamara Payment</a>	Tamara Configurations		5	<a href="#">View</a>

ID	Description	Sandbox	Production
tamaraEnablement	Enable the Tamara payment on storefront	Yes	Yes
tamaraEndPoint	Tamara API endpoint	<a href="https://api-sandbox.tamara.co/">https://api-sandbox.tamara.co/</a>	Merchant, please contact Tamara to get it.
tamaraAPIToken	API Token is used in every single request to Tamara API.	Merchant, please contact Tamara to get it.	Merchant, please contact Tamara to get it.

tamaraSupportedCountries	The list of Country Codes supports by Tamara	UAE, SA	UAE, SA
tamaraPriorityPDPWidget	Display Priority Tamara Widget in Product Page	PAY_BY_INSTALLMENTS	PAY_BY_INSTALLMENTS
tamaraNumberOfInstallments	Number of installments	3 (6)	3 (6)
tamaraWebhookEvents	Web Hook Events	["order_approved", "order_on_hold", "order_declined", "order_authorised", "order_canceled", "order_captured", "order_refunded", "order_expired"]	["order_approved", "order_on_hold", "order_declined", "order_authorised", "order_canceled", "order_captured", "order_refunded", "order_expired"]

#### 4.2.2 Service Credential

Go to **Administration > Operations > Services > Service Credentials**.

You will see all the services we are using.

[Administration](#) > [Operations](#) > [Services](#) > Service Credentials

Services Profiles **Credentials**

### Service Credentials

Select All	Name	URL	User
<input type="checkbox"/>	<a href="#">tamara.checkout.paymenttypes.cred</a>	/checkout/payment-types	
<input type="checkbox"/>	<a href="#">tamara.checkout.session.cred</a>	/checkout	
<input type="checkbox"/>	<a href="#">tamara.order.authorised.cred</a>	/orders/{orderId}/authorise	
<input type="checkbox"/>	<a href="#">tamara.order.cancel.cred</a>	/merchants/orders/{orderId}/cancel	
<input type="checkbox"/>	<a href="#">tamara.order.detail.cred</a>	/merchants/orders/{orderReferenceId}	
<input type="checkbox"/>	<a href="#">tamara.payment.capture.cred</a>	/merchants/orders/{orderId}/capture	
<input type="checkbox"/>	<a href="#">tamara.payment.refund.cred</a>	/merchants/orders/{orderId}/refund	
<input type="checkbox"/>	<a href="#">tamara.webhook.register.cred</a>	/webhooks	

Note that the User and password for each service should be empty

### 4.2.3 Order Failure Time

Go to Merchant Tools > Site Preferences > Order -> Find the Auto-Fail Orders setting, set it to 30 minutes

Payment Information Settings

Payment Information Retention:  Specifies the number of days after which the system will mask order payment information. Order payment instruments older than the specified number of days will be automatically masked so that the original credit card and bank account numbers are no longer recoverable. This masking is permanent and can't be undone should the number of days be changed later. Enter a value between 1 and 365.

Failed Orders Setting

Auto-Fail Orders: 30 Minutes after which the orders that haven't been placed become eligible for automatic failing. Leave this blank if non-placed orders should never be auto-failed.

Order Access Settings

Limit Storefront Order Access: No Specify how access to orders from the Storefront should be handled.

Storefront Order Filter by Customer Session: No When enabled, this adds an additional layer of security to order viewing in the Storefront, resulting in orders being filtered out which don't belong to the consumer identified in the Storefront session. The preference should be enabled unless there is a specific reason not to do it for a site.

### 4.2.4 Other considerations

As this Tamara payment is only support for some specific countries such as SA or UAE. Hence, your site should support the relevant locales, currencies, price to those countries.

*Please contact Tamara to know exactly the list of supported countries today.*

### 4.2.5 Register WebHook

Register the webhook notifications you receive to keep up to date with the status of your orders.

This process requires 1 time setting by running the controller TamaraHook-Register. This will call a Tamara web service to subscribe all the events mentioned in the “Web Hook Events”

1-7 of 7

Name	Value	Default Value
(tamaraSupportedCountries) The list of supported Country Code which is supported by Tamara	UAE SA	SA

The list of supported Country Code which is supported by Tam...

Web Hook Events

(tamaraWebhookEvents)  
Registered events when order status changed

[\"order\_approved\", \"order\_on\_hold\", \"order\_declined\", \"order\_authored\", \"order\_canceled\", \"order\_captured\", \"order\_refunded\", \"order\_expired\"]

Edit Across Sites

Registered events when order status changed



After finish the setup with all step above, please run TamaraHook-Register controller, you will receive the result with the “webhook\_id” as below

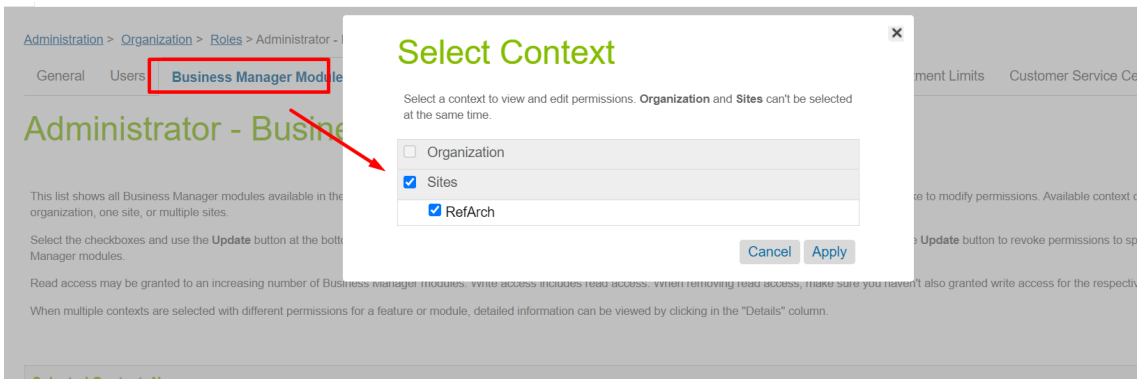
**Result**

```
{
  "webhook_id": "f67d497d-1d05-4b61-8cfe-fac9e5d96a3b",
  "url": "https://zzov-001.sandbox.us01.dx.commercecloud.salesforce.com/on/demandware.store/Sites-RefArch-Si"
}
```

Let record this ID somewhere in order to update webhook event in the future if you wish, following [the document here](#)

#### 4.2.6 Enable cartridge in Business Manager

From Administration > Organization > Roles >, select



## 5. Operation And Maintenance

### 5.1 Order Status Mapping

This section describes how the order status in Commerce Cloud is mapped with Tamara Order Status.

Tamara Order Status	SFCC Order Status	SFCC Payment Status	SFCC Shipping Status
approved	CREATED	NOT_PAID	NOT_SHIPPED
authorised	NEW/OPEN	NOT_PAID	NOT_SHIPPED
cancelled	CANCELLED	NOT_PAID	NOT_SHIPPED
captured	NEW/OPEN	PAID	SHIPPED
refunded	NEW/OPEN	NOT_PAID	SHIPPED

expired   declined	FAILED	NOT_PAID	NOT_SHIPPED
partial_cancelled	NEW/OPEN	PAID	NOT_SHIPPED
partial_refund	NEW/OPEN	PAID	SHIPPED

## 5.2 Order custom attributes

Some additional attributes are defined and stored in the Order object. These are the custom attributes and will be sent as a part of Order XML so that Order Management System can use the same for further processing.

Attribute ID	Type	Description
tamaraOrderReferenceID	String	Order ID on Tamara Payment
tamaraPaymentStatus	String	The payment status from Tamara
tamaraOrderDetail	String	The order information synced from Tamara for every action
tamaraCheckoutURL	String	The Checkout URL return back from Tamara Payment
tamaraSuccessNotification	String	The success notification URL when place order
tamaraFailureNotification	String	The failure notification URL when place order
tamaraCancelNotification	String	Tamara cancel notification URL when place order

## 5.3 TamaraPaymentTypes Custom Object

This object stores the information of Payment Types from Tamara which are utilized to display the available payment method and widget on the storefront.

Attribute ID	Type	Description
ID	String	ID of payment type

custom.content	String	Content of the payment type, this is a JSON object

## 5.4 Exception Log

Please get the related exception message related to Tamara Payment in the log file with the bellow pattern

```
custom-tamarapayment- <hostname> - appserver - <creation date of the file in GMT>.log
```

Possible Log	Description
total_amount_invalid_limit	The total amount is reached the min/max limit
consumer_empty_first_name	Missing the first name of customer
consumer_invalid_phone_number	Customer's phone number is invalid
consumer_invalid_email	Customer's email is invalid
consumer_empty_email	Customer's email is empty
total_amount_invalid_currency	We do not support cross currencies. Most likely the customer is buying from a non-supported country
not_supported_delivery_country	Not supported country

## 6. Question and Answer

Tamara to add the content here

Question	Answer

If you have any problems with implementing this API, please feel free to reach out to us at **(Tamara to provide the support contact)** and we will be more than happy to support you through the process.