Partner Integrations

Salesforce B2B Commerce Connector v1.1

The Salesforce B2B Commerce platform expertly solves B2B complexities and provides a smooth self-service shopping experience, while Digital River excels in providing solid ecommerce backend functionality. Rather than trying to handle all of these difficult issues, Digital River integrates with Salesforce to combine these best-of-breed strategies: Salesforce crafts a beautiful front-end purchasing experience, including product details, checkout, and payment information, while Digital River handles behind-the-scenes tax compliance, payment processing, and risk mitigation.

Using the Salesforce B2B Commerce Connector provides a host of benefits:

- Seamless connectivity with Digital River's financial backend
- Easy platform to install and configure
- No micromanagement of the front-end shopper experience
- Alleviates the burden of tax and fraud risk
- Unites best-in-class front-end and back-end solutions

This guide describes how the Salesforce B2B Commerce Connector works, how to install the connector, and how to customize it for your specific needs.

How it works

1. Use the Salesforce B2B Commerce Connector to enable Digital River to be the merchant of record for storefronts hosted in Salesforce B2B Commerce.
2. Salesforce B2B Commerce maintains all pricing and product data.
3. Digital River maintains minimal product data to fulfill the requirements for the merchant of record for tax calculations, tax collections, tax payments, and payment processing.
4. The CC Product in Salesforce B2B Commerce is automatically synced to Digital River with minimal information needed to calculate taxes and enable the merchant of record functions.
5. Digital River maintains a copy of the cart, customer, and order.

Integration points

1. Tax Estimations

1. The connector extends the global extension point `ccrz.cc_hk_TaxCalculation`.
2. The extension creates a Digital River cart based on the Salesforce B2B Commerce cart with product information, prices, and quantity. This cart is serialized as a JSON object, encrypted, and posted to a Digital River web service.
3. The response contains the cart tax amount.

2. Payment Processing

You can integrate payment in multiple locations and flows.

During checkout

In the standard checkout flow, you can apply the payment against the order using one of the following methods:

- Credit Card
- PayPal
- Purchase Order
- Previously stored payment method

My Wallet Page

- You can add new payment methods and update existing payment methods.
- Only credit card payment methods support saved payment options.

Installation requirements

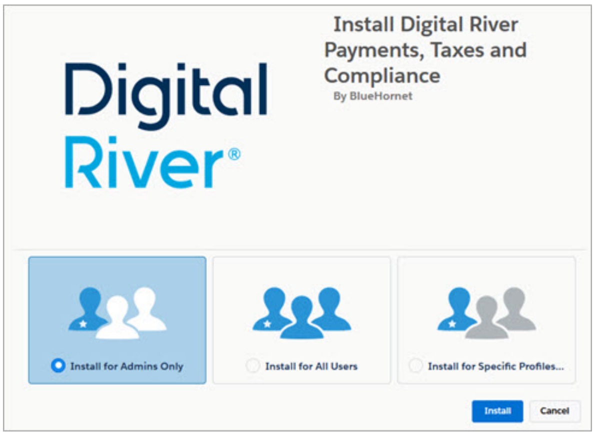
- You can install the Salesforce B2B Commerce Connector only on the Enterprise and Unlimited versions of Salesforce.
- B2B Commerce should be already installed on the system.

- Communities should be enabled in the Salesforce version before you install this application.
- You should assign a role to the community users (Customers).
- Before you click Install, you must map your B2B Customer profile to provide the appropriate access and permissions.

Step 1. Install the Salesforce B2B Commerce Connector

To install the Salesforce B2B Commerce Connector:

1. Run the installation from the AppExchange.
2. Go to the installation to open the Installation page of the application.

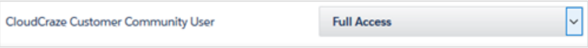


Install Digital River Payments, Taxes and Compliance
By BlueHornet

● Install for Admins Only
○ Install for All Users
○ Install for Specific Profiles...

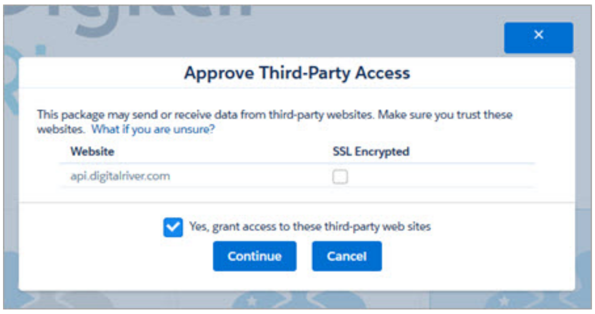
3. Click Install for Specific Profiles.

Important: Before you click Install, you must map your B2B Customer profile to provide the appropriate access and permissions.



CloudCraze Customer Community User: Full Access

4. Click the Install button. The Approve Third-Party Access page appears.



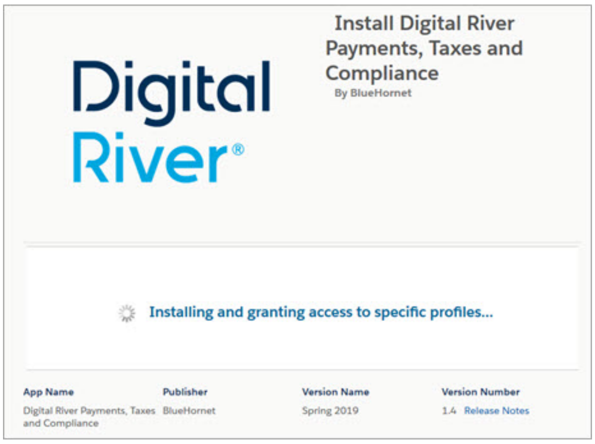
Approve Third-Party Access

This package may send or receive data from third-party websites. Make sure you trust these websites. What if you are unsure?

Website	SSL Encrypted
api.digitalriver.com	□

☑ Yes, grant access to these third-party web sites

5. Select the Grant Access check box and click the Continue button to start the installation process.



Install Digital River Payments, Taxes and Compliance
By BlueHornet

Installing and granting access to specific profiles...

App Name	Publisher	Version Name	Version Number
Digital River Payments, Taxes and Compliance	BlueHornet	Spring 2019	1.4 Release Notes

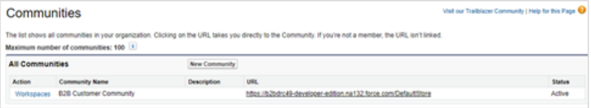
The installation process will take approximately 5-10 minutes. When the installation completes, the user who initiated the install will receive an email.

Step 2. Enable the Salesforce B2B Commerce Connector Config page

After you have received the Successful Installation confirmation via email, you need to configure the application.

To enable the digitalriverb2b.DRConnectorConfig page:

1. Go to Salesforce Setup, select Customize, select Communities, and then click All Communities.
2. Open the Community Management page of the B2B Commerce Cloud in your organization.



Communities

The list shows all communities in your organization. Clicking on the URL takes you directly to the Community. If you're not a member, the URL isn't linked.

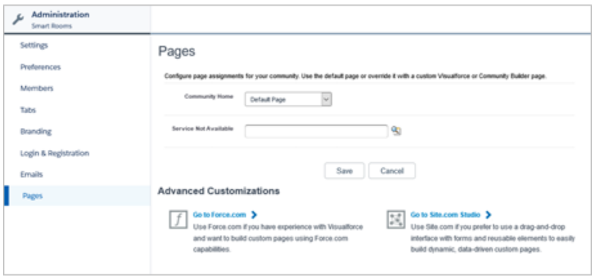
Maximum number of communities: 100

All Communities

Action	Community Name	Description	URL	Status
Workspaces	B2B Customer Community		https://b2bdrc49-developer-edition.na132.force.com/DefaultStore	Active

3. On the Community Management screen, click the Administration box to open the Administration Management of the Community.

4. Click Pages in the navigation menu on the left and then click the Go to Force.com link.



Pages

Configure page assignments for your community. Use the default page or override it with a custom Visualforce or Community Builder page.

Community Home: Default Page

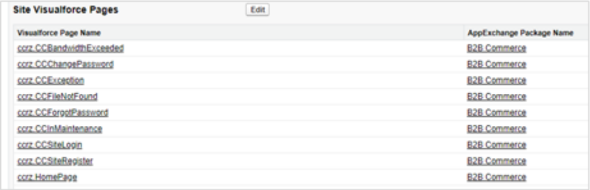
Service Not Available:

Advanced Customizations

Go to Force.com
Use Force.com if you have experience with Visualforce and want to build custom pages using Force.com capabilities.

Go to Site.com Studio
Use Site.com if you prefer to use a drag-and-drop interface with forms and reusable elements to easily build dynamic, data-driven custom pages.

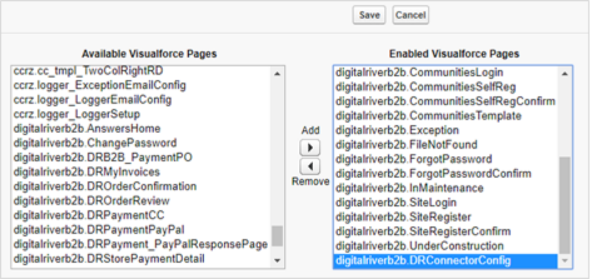
5. On the Site Details page, scroll down to the Site Visualforce Pages and click the Edit button.



Site Visualforce Pages

Visualforce Page Name	AppExchange Package Name
ccrz.CCBandwidthExceeded	B2B Commerce
ccrz.CCChangePassword	B2B Commerce
ccrz.CCException	B2B Commerce
ccrz.CCFileNotFound	B2B Commerce
ccrz.CCForgotPassword	B2B Commerce
ccrz.CCInMaintenance	B2B Commerce
ccrz.CCSiteLogin	B2B Commerce
ccrz.CCSiteRegister	B2B Commerce
ccrz.HomePage	B2B Commerce

6. Add the digitalriverb2b.DRConnectorConfig page to the Enabled Visualforce Pages and click the Save button.



Available Visualforce Pages

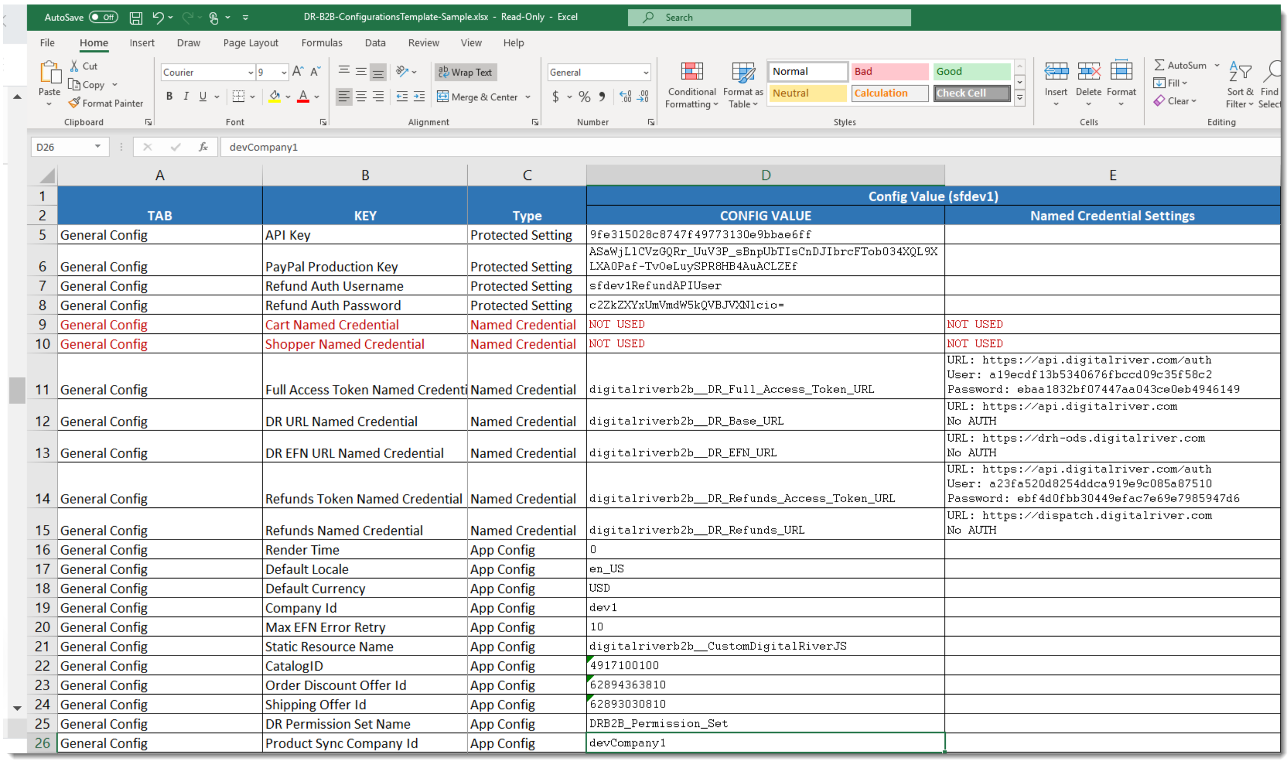
ccrz.cc_tmpl_TwoColRightRD
ccrz.logger_ExceptionEmailConfig
ccrz.logger_LoggerEmailConfig
ccrz.logger_LoggerSetup
digitalriverb2b.AnswersHome
digitalriverb2b.ChangePassword
digitalriverb2b.DRB2B_PaymentPO
digitalriverb2b.DRMyInvoices
digitalriverb2b.DROrderConfirmation
digitalriverb2b.DROrderReview
digitalriverb2b.DRPaymentCC
digitalriverb2b.DRPaymentPayPal
digitalriverb2b.DRPayment_PayPalResponsePage
digitalriverb2b.DRStorePaymentDetail

Enabled Visualforce Pages

digitalriverb2b.CommunitiesLogin
digitalriverb2b.CommunitiesSelfReg
digitalriverb2b.CommunitiesSelfRegConfirm
digitalriverb2b.CommunitiesTemplate
digitalriverb2b.Exception
digitalriverb2b.FileNotFound
digitalriverb2b.ForgotPassword
digitalriverb2b.ForgotPasswordConfirm
digitalriverb2b.InMaintenance
digitalriverb2b.SiteLogin
digitalriverb2b.SiteRegister
digitalriverb2b.SiteRegisterConfirm
digitalriverb2b.UnderConstruction
digitalriverb2b.DRConnectorConfig

Step 3. Configure the Salesforce B2B Commerce Connector

To configure the Salesforce B2B Commerce Connector from the All Tabs Menu, click the DR Connector Configure tab and complete the following steps. To complete the configurations, you will need information about API keys and other details that Digital River will provide to you in the DR B2B Configurations Template spreadsheet. The following image shows a sample of the template.



	A	B	C	D	E
1				Config Value (sfdev1)	
2	TAB	KEY	Type	CONFIG VALUE	Named Credential Settings
5	General Config	API Key	Protected Setting	9fe315028c8747f49773130e9bbae6ff	
6	General Config	PayPal Production Key	Protected Setting	ASaWjL1CVzGQRr_UuV3P_sBnpUbTIsCnDJIbrcFTob034XQL9X LXA0Paf-TvOeLuySPR8HB4AuACLZEf	
7	General Config	Refund Auth Username	Protected Setting	sfdev1RefundAPIUser	
8	General Config	Refund Auth Password	Protected Setting	c2ZkZXYxUmVmdW5kQVBJVXNlcio=	
9	General Config	Cart Named Credential	Named Credential	NOT USED	NOT USED
10	General Config	Shopper Named Credential	Named Credential	NOT USED	NOT USED
11	General Config	Full Access Token Named Credenti	Named Credential	digitalriverb2b__DR_Full_Access_Token_URL	URL: https://api.digitalriver.com/auth User: a19ecdf13b5340676fbccd09c35f58c2 Password: ebaa1832bf07447aa043ce0eb4946149
12	General Config	DR URL Named Credential	Named Credential	digitalriverb2b__DR_Base_URL	URL: https://api.digitalriver.com No AUTH
13	General Config	DR EFN URL Named Credential	Named Credential	digitalriverb2b__DR_EFN_URL	URL: https://drh-ods.digitalriver.com No AUTH
14	General Config	Refunds Token Named Credential	Named Credential	digitalriverb2b__DR_Refunds_Access_Token_URL	URL: https://api.digitalriver.com/auth User: a23fa520d8254ddca919e9c085a87510 Password: ebf4d0fbb30449efac7e69e7985947d6
15	General Config	Refunds Named Credential	Named Credential	digitalriverb2b__DR_Refunds_URL	URL: https://dispatch.digitalriver.com No AUTH
16	General Config	Render Time	App Config	0	
17	General Config	Default Locale	App Config	en_US	
18	General Config	Default Currency	App Config	USD	
19	General Config	Company Id	App Config	dev1	
20	General Config	Max EFN Error Retry	App Config	10	
21	General Config	Static Resource Name	App Config	digitalriverb2b__CustomDigitalRiverJS	
22	General Config	CatalogID	App Config	4917100100	
23	General Config	Order Discount Offer Id	App Config	62894363810	
24	General Config	Shipping Offer Id	App Config	62893030810	
25	General Config	DR Permission Set Name	App Config	DRB2B_Permission_Set	
26	General Config	Product Sync Company Id	App Config	devCompany1	

General Configuration

The General Config tab provides information that is used by the connector to establish secure connections between the Salesforce B2B Commerce platform and Digital River. This tab also includes general information about the Digital River site and catalogs linked to the storefront. To complete this step, you will need the following information from Digital River to configure your account.

- The encryption algorithm and encryption key
- API key
- User ID and password
- The default locale
- Default currency
- Company ID

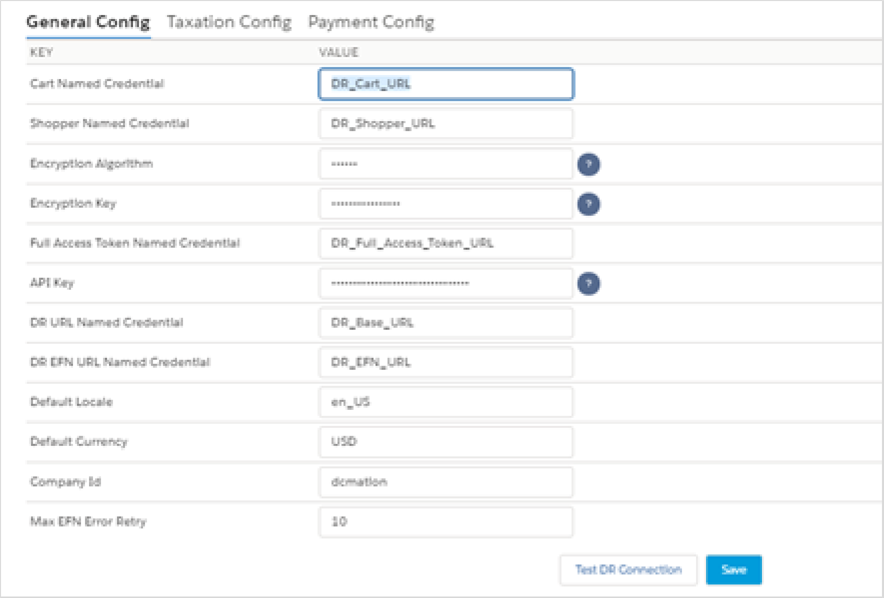
Update the named credentials

The managed package installs the following Named credentials that the Salesforce B2B Commerce Connector uses to communicate with the Digital River web services.

- Full Access Token Named Credential

- DR URL Named Credential
- DR EFN URL Named Credential
- Refunds Token Named Credential
- Refunds Named Credential

You can update the named credentials with the details provided in the spreadsheet, or create your own named credentials and use them in the configuration tabs.



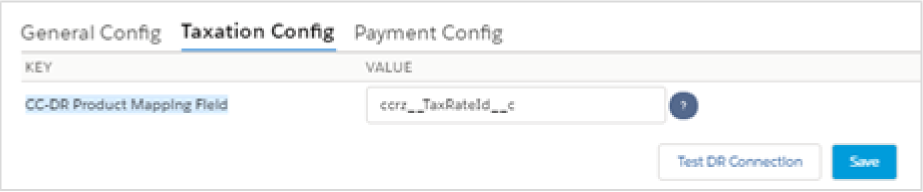
KEY	VALUE
Cart Named Credential	DR_Cart_URL
Shopper Named Credential	DR_Shopper_URL
Encryption Algorithm	••••••
Encryption Key	•••••••••••••••••
Full Access Token Named Credential	DR_Full_Access_Token_URL
API Key	••••••••••••••••••••••••••••••••••
DR URL Named Credential	DR_Base_URL
DR EFN URL Named Credential	DR_EFN_URL
Default Locale	en_US
Default Currency	USD
Company Id	dcmation
Max EFN Error Retry	10

General Config

Note: Click the Test DR Connection button to verify whether you can successfully connect with the configured Digital River site.

Taxation Configuration

As of version 1.9 of the Connector, the Taxation Config tab is no longer used. The Product Sync tab replaces this functionality and automatically synchronizes the Commerce Connector Products in Salesforce to Digital River along with applicable tax codes and ECCNs. The setting on this tab does not affect the functioning of the connector. The tab deprecated will be removed in a future version.



KEY	VALUE
CC-DR Product Mapping Field	ccrz__TaxRateId__c

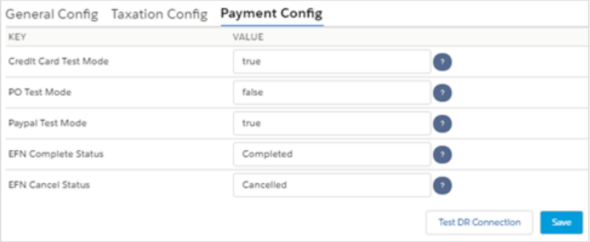
Taxation Config

Payment Configuration

Use the Payment Config tab to enter the payment details from the Digital River API.

1. Set the following Payment Test modes under the Payment Config tab to True in the development and test environments and False in the production environment:
   - Credit Cart Test Mode
   - PO Test Mode
   - PayPal Test Mode
   - PO Test Mode

   Note: When you set these modes to True, the Cart is marked as in Test while sending calls to Digital River and prevents the charging of credit cards and other payment methods.
2. Set the EFN Complete Status to Completed. The Salesforce B2B Commerce Connector sends an Electronic Fulfillment Notification (EFN) to Digital River when the order is fulfilled to begin the capture of the funds. Since the status of the order on fulfillment is implementation-dependent, this property allows the implementation to override the default status that triggers the notification.
3. Set the EFN Cancel Status to Cancelled. The Salesforce B2B Commerce Connector sends an Electronic Fulfillment Notification (EFN) to Digital River when the order is canceled to begin a refund of the funds. Since the status of the order cancellation is implementation-dependent, this property allows the implementation to override the default status that triggers the notification.

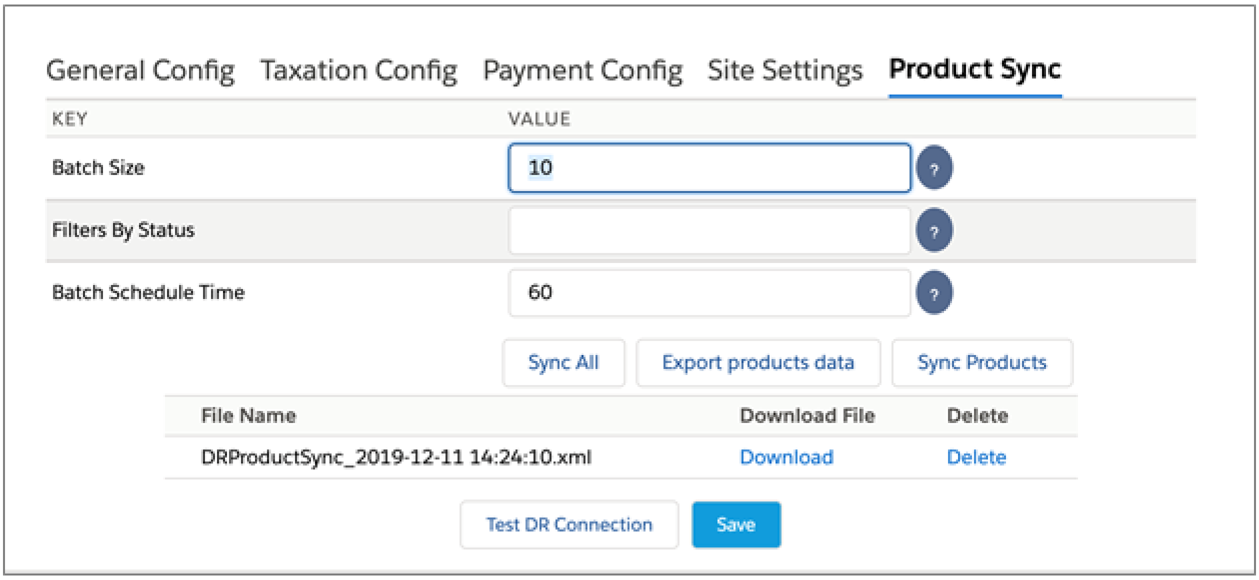
   

KEY	VALUE
Credit Card Test Mode	true
PO Test Mode	false
Paypal Test Mode	true
EFN Complete Status	Completed
EFN Cancel Status	Cancelled
4. Click Save.

Step 4. Configure Product Sync

The Product Sync page allows you to configure the behavior of the automated product synchronization. A Batch job performs the synchronization, and you can schedule it to run periodically.

1. Under the Product Sync tab, configure the following attributes:
   - **Size**: This is the number of CC product records that will be pushed to Digital River in each run of the job.
   - **Filter By Status**: By default, all products with a status of ‘Released’ are synced. Add a comma-separated list of CC Product statuses to override this behavior.
   - **Batch Schedule Time**: Defines the frequency of the batch job run in minutes.
2. Click Save.



KEY	VALUE
Batch Size	10
Filters By Status	
Batch Schedule Time	60

File Name	Download File	Delete
DRProductSync_2019-12-11 14:24:10.xml	Download	Delete

Step 5. Create page labels

The Page Labels are required to show and make the payments via Credit Card, PayPal, and Purchase Orders using Digital River Connector.

Open the CC Labels tab from the All Tabs menu and create the following Global Page Labels:

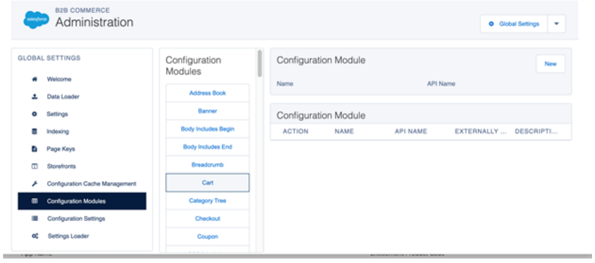
	Page Name	Page Label ID	Page Label Name
1	All	PaymentDisplay_ccGlobalAll	PaymentDisplay_
2	All	MyWallet_AcctType_ccGlobalAll	MyWallet_AcctTy
3	All	PaymentProcessorTab_ccGlobalAll	PaymentProcesso
4	All	MyWallet_AcctNum_cc	MyWallet_AcctNu
5	MyAccount	PaymentDisplay_ppGlobalMyAccount	PaymentDisplay_
6	OrderView	PaymentDisplay_ppGlobalOrderView	PaymentDisplay_
7	CheckOutNew	PaymentDisplay_ppGlobalCheckOutNew	PaymentDisplay_
8	OrderConfirmation	PaymentDisplay_ppGlobalOrderConfirmation	PaymentDisplay_
9	All	MyWallet_AcctNum_ppGlobalAll	MyWallet_AcctNu
10	All	MyWallet_AcctType_ppGlobalAll	MyWallet_AcctTy
11	All	PaymentProcessorTab_ppGlobalAll	PaymentProcesso

Step 6. Configure CC Admin settings

You need to override several settings to enable the custom payment methods and tax calculations. The various settings are detailed in the spreadsheet accompanying this document. The appropriate tabs are referenced in the following sections.

Step 6a. Create new Configuration Modules for Payment types

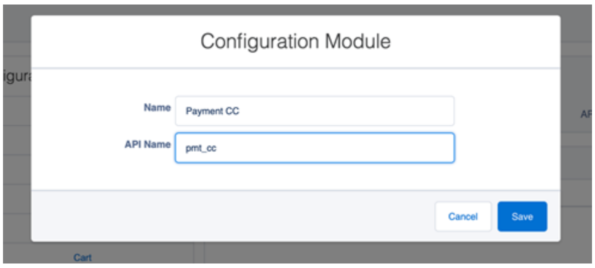
1. Select CC Admin and then select Configuration Modules as shown below.



2. Create a module for Credit Cards.

   a. Click New in the Configuration Module pane. The Configuration Module dialog appears.

   b. Enter `Payment CC` in the Name field and `pmt_cc` in the API Name field.



Configuration Module

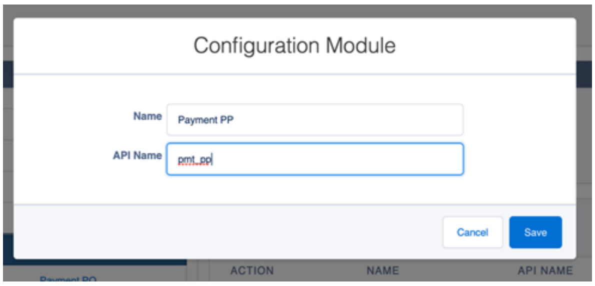
Name: Payment CC

API Name: pmt_cc

3. Create a module for PayPal.

   a. Click New in the Configuration Module pane.

   b. Enter `Payment PP` in the Name field and `pmt_pp` in the API Name field.

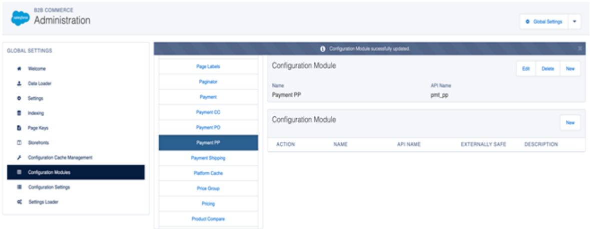


Configuration Module

Name: Payment PP

API Name: pmt_pp

The new payment modules will appear under Configuration Modules.



Configuration Module sucessfully updated.

Configuration Module

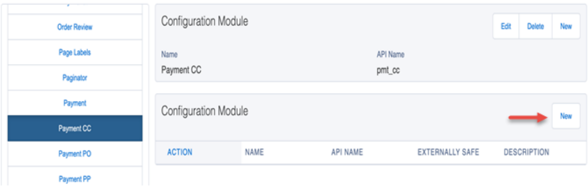
Name	API Name
Payment PP	pmt_pp

Step 6b. Create new configuration metadata for the Payment Modules

1. Create a new configuration metadata for the Payment CC module.

   a. Under Configuration Modules, click Payment CC.

   b. Click New (highlighted in the image below) to create new configuration metadata for the Payment CC module. The Configuration Metadata dialog appears.



Configuration Module

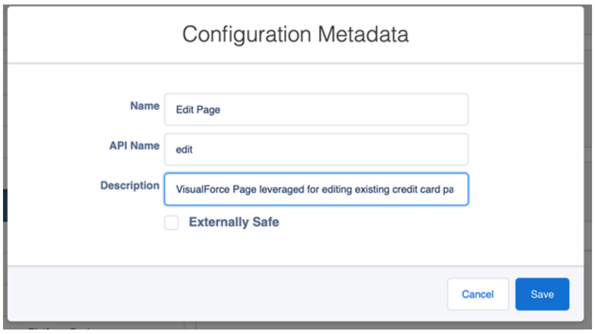
Name: Payment CC

API Name: pmt_cc

Configuration Module

ACTION	NAME	API NAME	EXTERNALLY SAFE	DESCRIPTION

c. Complete the fields to create configuration metadata for the Edit Page. The following image shows an example of the configuration metadata for the Edit Page.



Configuration Metadata

Name: Edit Page

API Name: edit

Description: VisualForce Page leveraged for editing existing credit card pa

□ Externally Safe

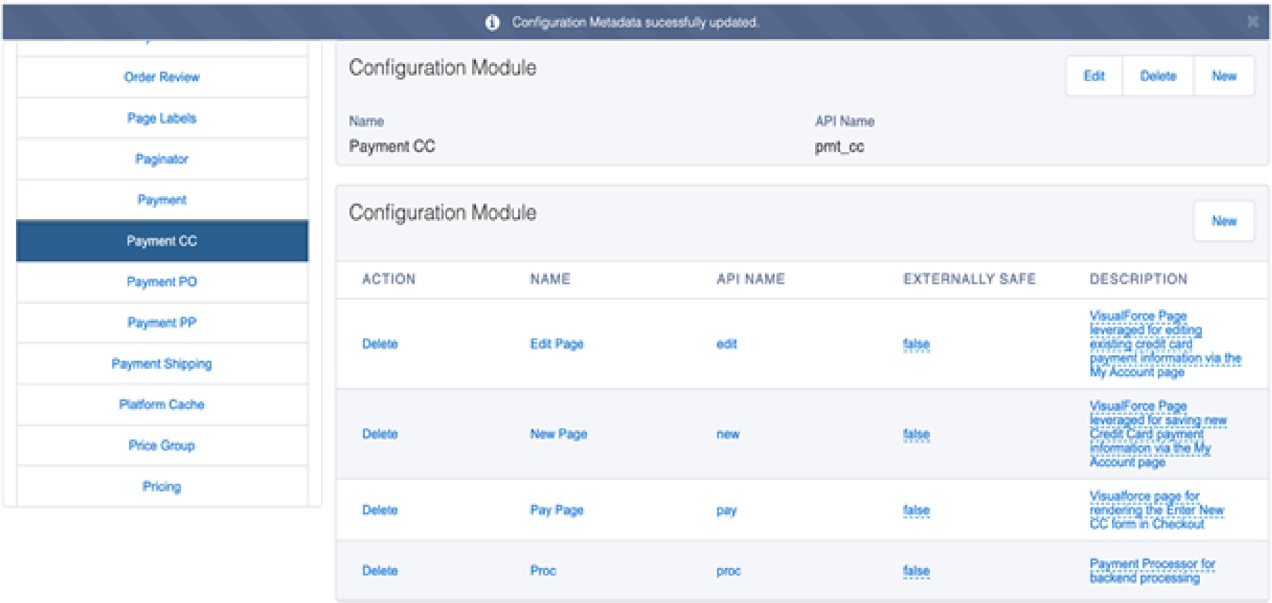
d. Repeat steps b and c to create configuration metadata for Pay Page, New Page, and Proc.

2. Create a new configuration metadata for the Payment PP module.

   a. Under Configuration Modules, click Payment PP.

   b. In step 4, repeat steps c through d for the Payment PP module.

After adding the configuration metadata to both of the Payment Modules, the Configuration Module for both modules should look like:



Configuration Metadata sucessfully updated.

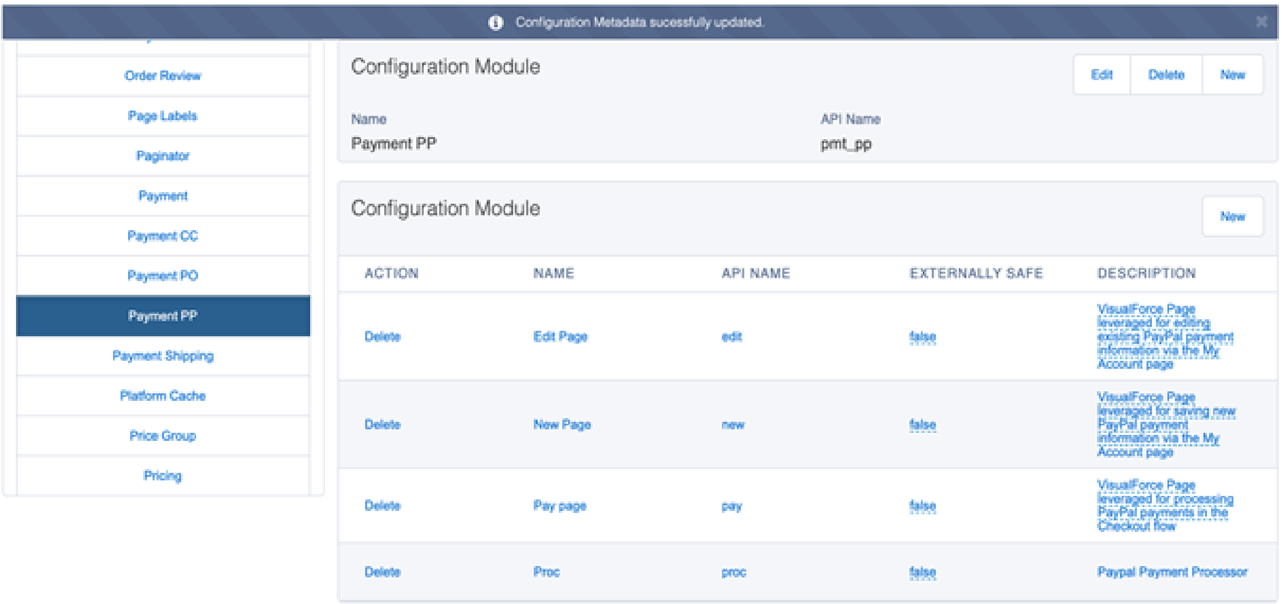
Configuration Module

Name: Payment CC

API Name: pmt_cc

Configuration Module

ACTION	NAME	API NAME	EXTERNALLY SAFE	DESCRIPTION
Delete	Edit Page	edit	false	VisualForce Page leveraged for editing existing credit card payment information via the My Account page
Delete	New Page	new	false	VisualForce Page leveraged for saving new Credit Card payment information via the My Account page
Delete	Pay Page	pay	false	Visualforce page for rendering the Enter New CC form in Checkout
Delete	Proc	proc	false	Payment Processor for backend processing



Configuration Module

Name: Payment PP

API Name: pmt_pp

Configuration Module

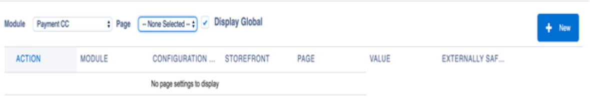
ACTION	NAME	API NAME	EXTERNALLY SAFE	DESCRIPTION
Delete	Edit Page	edit	false	VisualForce Page leveraged for editing existing PayPal payment information via the My Account page
Delete	New Page	new	false	VisualForce Page leveraged for saving new PayPal payment information via the My Account page
Delete	Pay page	pay	false	VisualForce Page leveraged for processing PayPal payments in the Checkout flow
Delete	Proc	proc	false	Paypal Payment Processor

Step 6c. Create or update configuration settings

1. To create or update the configuration settings:

   a. From CC Admin, select the storefront you want to configure and then select Configuration Settings.

   b. Select Payment CC from the Module dropdown list.

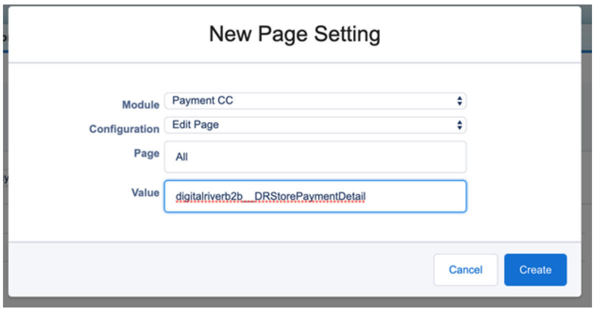


Module: Payment CC   Page: -- None Selected --   ✓ Display Global

ACTION	MODULE	CONFIGURATION ...	STOREFRONT	PAGE	VALUE	EXTERNALLY SAF...
No page settings to display						

   c. To create a new Page Setting for the module, click New.

   d. Complete the fields to create or update the page settings. The following image shows an example of the page settings for the Edit Page for the Payment CC module.



New Page Setting

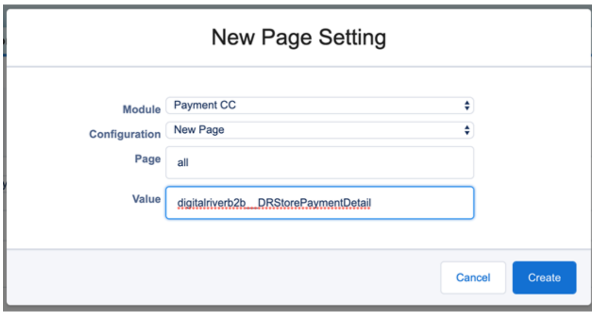
Module: Payment CC

Configuration: Edit Page

Page: All

Value: digitalriverb2b__DRStorePaymentDetail

   e. Repeat step b through d for New Page, Pay Page, and Proc.



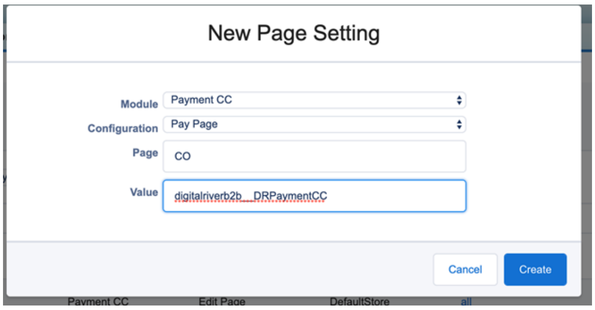
New Page Setting

Module: Payment CC

Configuration: New Page

Page: all

Value: digitalriverb2b__DRStorePaymentDetail



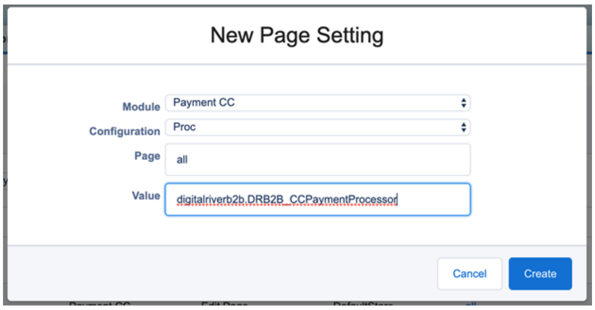
New Page Setting

Module: Payment CC

Configuration: Pay Page

Page: CO

Value: digitalriverb2b__DRPaymentCC



New Page Setting

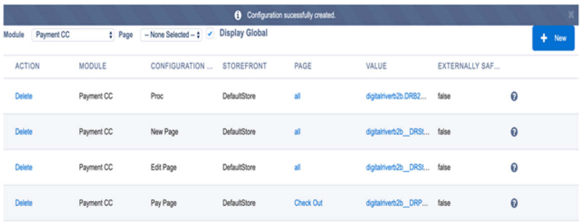
Module: Payment CC

Configuration: Proc

Page: all

Value: digitalriverb2b.DRB2B_CCPaymentProcessor

After creating or updating the configuration settings for the Payment CC module, the Payment CC Module should look like:



Configuration sucessfully created.

Module: Payment CC Page: -- None Selected -- ☑ Display Global

ACTION	MODULE	CONFIGURATION ...	STOREFRONT	PAGE	VALUE	EXTERNALLY SAF...
Delete	Payment CC	Proc	DefaultStore	all	digitalriverb2b.DRB2...	false
Delete	Payment CC	New Page	DefaultStore	all	digitalriverb2b__DRSt...	false
Delete	Payment CC	Edit Page	DefaultStore	all	digitalriverb2b__DRSt...	false
Delete	Payment CC	Pay Page	DefaultStore	Check Out	digitalriverb2b__DRP...	false

f. Repeat steps b through e for each additional payment module.

Step 7. Schedule backend jobs

For the connector to continuously synchronize data between Salesforce and Digital River, you need to schedule several backend jobs. To schedule the backend jobs.

1. Open an Execute Anonymous window in the Developer Console or the salesforce workbench.

2. Copy the script in each row in the table into the window and execute the scripts one after another to schedule all the jobs.

The following table lists the jobs that you need to schedule and the apex code that you can run to set up the jobs.

Job Name	Script for Scheduled Job

Order Level Electronic Fulfillment Job	// EFN To run every 5 minute for(Integer i=0; i<60; i=i+5){
	  String cronTrigger = '0 ' + i + ' * ?';
	  String jobID = System.schedule('Job Send EFN every 5 minutes' + i, cronTrigger, new digitalriverb2b.DRB2B_ScheduleOrderLevelEFN());
	  System.debug('Scheduled Job ID: ' + jobID);
	}

Line Item Electronic Fulfillment Job	// Line Item EFN every 5 minutes for(Integer i=0; i<60; i=i+5){
	  String cronTrigger = '0 ' + i + ' * ?';
	  String jobID = System.schedule('Job Send Line Item EFN every 5 minutes' + i, cronTrigger, new digitalriverb2b.DRB2B_ScheduleLineItemEFN());
	  System.debug('Scheduled Job ID: ' + jobID);
	}

Merge Duplicate Fulfillment Records	// Merge Duplicate Fulfillments to run every minute for(Integer i=0; i<60; i=i+1){
	  String cronTrigger = '0 ' + i + ' * ?';
	  String jobID = System.schedule('Merge Duplicate Fulfillment every minute' + i, cronTrigger, new digitalriverb2b.DRB2B_MergeDuplicateFulfillment());
	  System.debug('Scheduled Job ID: ' + jobID);
	}

CC Subscription Processor Job	// Schedule CC Subscription Processor Job
	System.schedule('CC Subscription Procesor Job Hourly', '0 0 * ?', new digitalriverb2b.DRB2B_ScheduledSubscriptionProcess());

DR Installament Subscription Processor Job	//Schedule Installment Subscription Processor Job every hour
	System.schedule('DR Installment Subscription Processor Job Hourly', '0 0 * ?', new digitalriverb2b.DRB2B_InstallmentSubscriptionProcessor());