



---

# Link Cart to Cash

Salesforce, Spring '24





# CONTENTS

- [Link Cart to Cash](#) ..... 1
- Workflow ..... 2
- Design Considerations ..... 3
- Connector ..... 4
- Configurations ..... 8



# LINK CART TO CASH

Connect B2B Commerce with CPQ billing and order information to improve the revenue management process.



## Get Started

---



Take Trailhead modules related to this solution.

- [Salesforce Solution Kits: Quick Look](#)

This solution kit helps you:

- Use a connector to synchronize transactions from cart to cash with minimal development.
- Connect data flows for billing and order orchestration.
- Collect transaction data across clouds to improve retention.
- Empower marketing, sales, and support teams by improving subscription payments and processing.

## Required Products

---

- B2B Commerce Cloud
- Salesforce CPQ
- Sales Cloud, Service Cloud, or an Industry Cloud such as Consumer Goods Cloud or Manufacturing Cloud
- B2B Commerce and CPQ Connector

## Implement This Solution

---

### [Workflow](#)

Learn how data flows through the configurations to link cart to cash.

### [Design Considerations](#)

For the best outcomes, keep these design considerations in mind. Links and other resources help you link cart to cash.

### [Connector](#)

Connectors are developer enablement frameworks that accelerate cross-cloud integration by providing code, configuration, and implementation patterns. Use the CPQ B2B Commerce Cloud Connector to link cart to cash.

[Configurations](#)

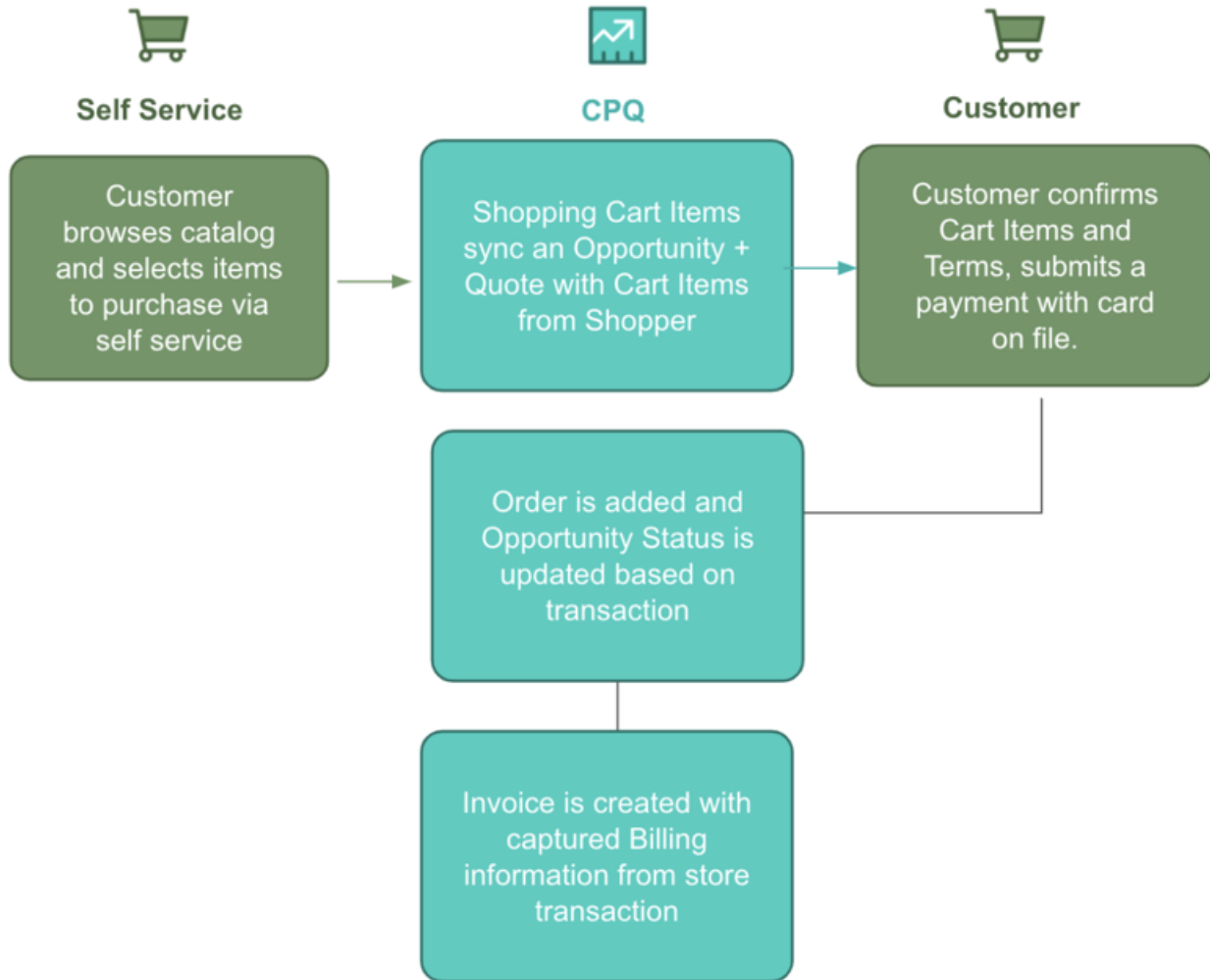
Use these configurations to link cart to cash.

## Workflow

---

Learn how data flows through the configurations to link cart to cash.

### Workflow



### Related Content



Take the next steps in this implementation.

- [Design Considerations](#)
- [Connector](#)

- [Configurations](#)

## Design Considerations

---

For the best outcomes, keep these design considerations in mind. Links and other resources help you link cart to cash.

### Best Practices

#### Invoice Integration Between B2B and Billing

- We recommend that you directly surface the billing invoices in the B2B Storefront, rather than synchronize the invoices between B2B Invoice and Billing Invoice objects.
- Use a custom component that surfaces the invoices and payments directly from Salesforce Billing to the B2B storefront.

#### Billing-Related Data Capture at Sign Up or Check Out

- Frequency, Method
- Billing contact info
- All subscription (terms, length, and so on) at sign up or check out

#### Invoice Integration Between B2B and Billing

Use the B2B Commerce data model flow “CC Invoice” and “CC Payment” objects to capture the Invoice and Payment details.

#### Salesforce Billing Fields

We recommend that you don’t synchronize the “CC Invoice” and “CC Payment” objects with “Invoice” and “Payment” objects in Salesforce Billing.

### Key Considerations

#### Cybersource Marketplace Template

If you’re using the Cybersource marketplace template available on the out-of-box B2B storefront, set the “Auth and Capture” value to “False”.

#### Storefront Guest User Access Orders

If the storefront allows unregistered users to check out, add the necessary Apex class permissions to your community’s guest user profile.

### Performance and Scalability Considerations

#### ERP Integrations

- This model lets the user extend functionality to an ERP or any backend system for enhanced revenue recognition or fulfillment.
- When activated, the Order object is the trigger point when sending any order information to an external system. It’s also the entry point back when updating any fulfillment-level details.
- Implement the Order-to-Cash (OTC) integration between Salesforce Billing and the underlying ERP. It’s the single source of truth for orders, invoices, and electronic payments regardless of whether it originated in B2B or CPQ.
- Integrating OTC with B2B isn’t necessary.

#### Tax Integration

If using an external tax engine like Avalara, Vertex, or Sabrix, integrate B2B and Billing with the tax engine to avoid any discrepancies.

### Pricing

Whether the order amounts come from a B2B (self-service) or CPQ (sales rep) transaction, ensure that the orders match.

- We recommend that you set the rounding precision to two decimal places. This precision must be the same in both B2B and CPQ.
- A self-service B2B order automatically creates CPQ Orders. The net price from the B2B Order publishes and discounts as the “Special Price” and “Net Price” on the CPQ Quote Lines.

Ensure that any price rules in CPQ ignore the B2B system-generated quotes as part of the self-service flow. Track it by adding a custom attribute on the Quoting object. For example, “Quote\_Source\_\_c” B2B. Check for this value in all Price Rules or Quote Calculator Plugin (QCP) customizations.

### Billing and Charge Types

Set up the Billing and Charge types to have the same values between B2B and CPQ for any supported products that use these cross-cloud scenarios.

## See Also

- [CPQ | B2B Connector Setup and Implementation Guide](#)
- [B2B Commerce Data Model](#)
- [Salesforce Billing Hosted Payment Page](#)
- [CyberSource for B2B Commerce](#)
- [Salesforce Order Data Model](#)

## Related Content



Review earlier steps in this solution.

- [Workflow](#)



Take the next steps in this implementation.

- [Connector](#)
- [Configurations](#)

## Connector

---

Connectors are developer enablement frameworks that accelerate cross-cloud integration by providing code, configuration, and implementation patterns. Use the CPQ B2B Commerce Cloud Connector to link cart to cash.

The CPQ B2B Commerce Cloud Connector is an unmanaged package from Salesforce Labs. It allows B2B Commerce and CPQ customers to configure or customize functionality. Sync products, pricing, quote requests, and orders in both clouds.

Use one data model, pricing engine, and a unified process built around customer engagement from CPQ. Create Omni-Channel customer-driven experiences.

The connector supports the following use cases:

1. Synchronize Product and Price Data



2. Convert Cart to Order
3. Request for a Quote
4. Buy Again with Synchronized Order History


## General Connector Information

- Connectors are a developer enablement framework that accelerates cross-cloud integration. Connectors aren't a standalone product.
- Connectors support a core set of use cases and can be extended to support more customer-driven use cases.
- Connectors require customization and configuration in B2B Commerce and Salesforce CPQ.
- Plan your connector implementation with a data strategy and governance policy.

## What Your Company Can Do with This Connector

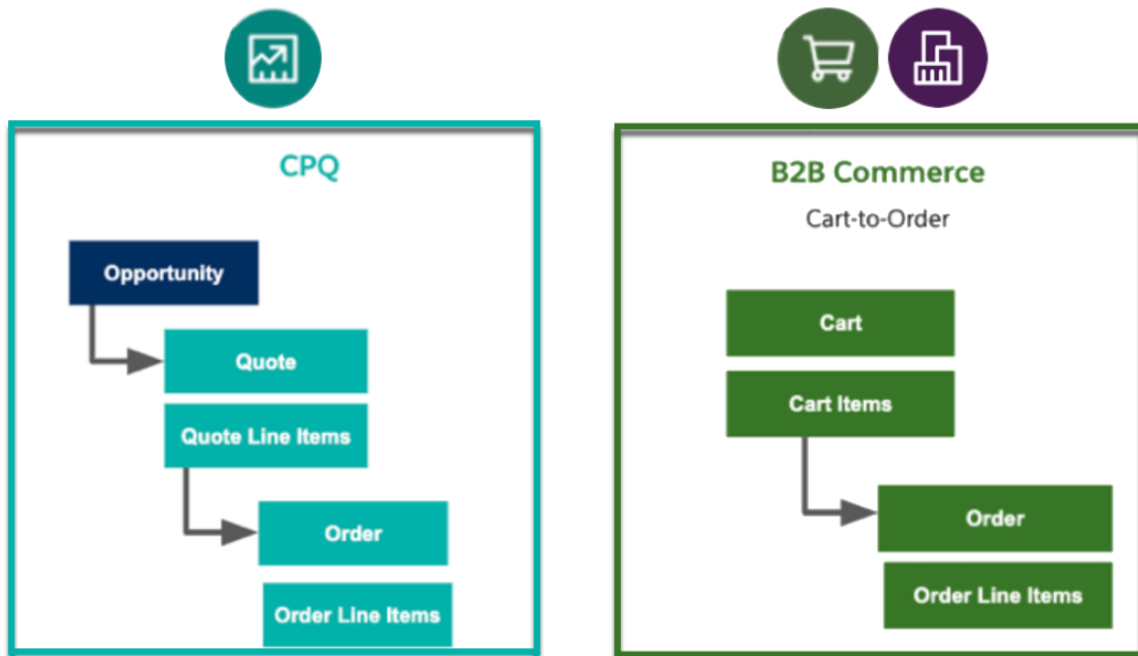
Each connection point supports a core set of use cases enabled individually through custom metadata configurations. These connections meet your specific business requirements. You can disable them as needed, such as for scheduled maintenance or data ETL activities.

- Product and Price book
- Cart and Quoting
- Orders and CC Order

 **Note:** See the following visualized data flows. You reduce system complexity with a unified master data, shared data model, and single source of truth.

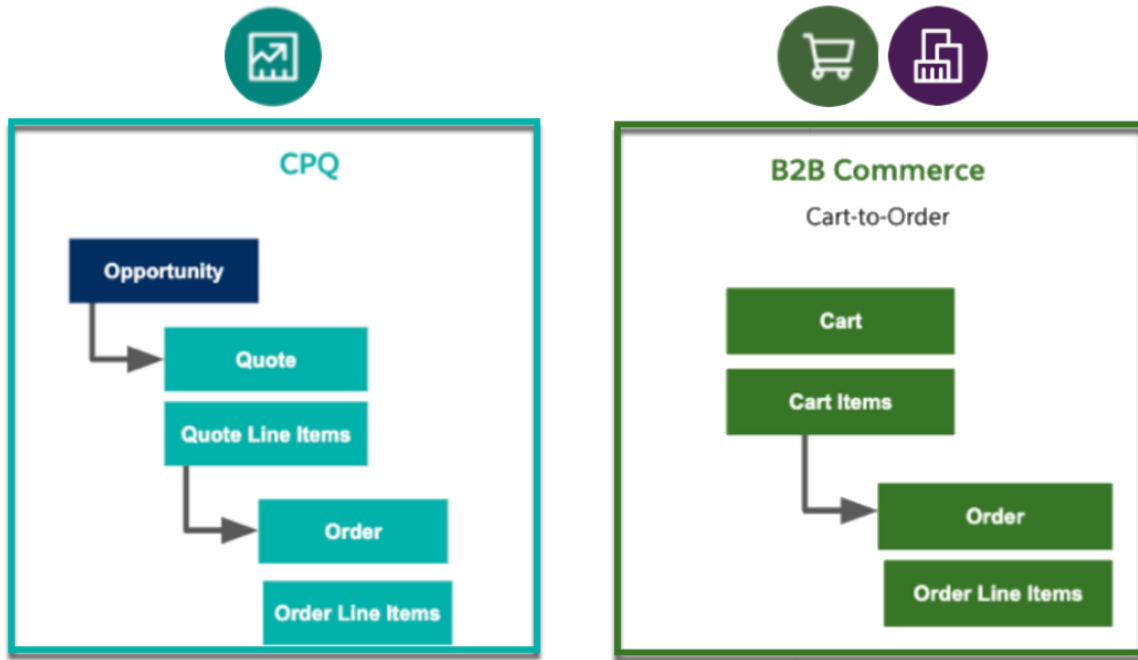
### Connected Selling and Buying

Synchronize Quoting and Order History.



**Connected Products and Pricing**

Create CPQ Product and Price Data and Sync to B2B Commerce.



**Note:** The CPQ + B2B Connector supports a one-way sync for Product and Pricing logic.

**Table 1: Data Synchronization**

Data Sync	Approach
Products and Pricing	<ul style="list-style-type: none"> <li>• Platform Events</li> <li>• Scheduled Batch Classes</li> <li>• Queue-able Framework</li> </ul>
Transactional Carts, Quotes, and Orders	<ul style="list-style-type: none"> <li>• Triggers</li> <li>• Future Methods</li> </ul>

**Factors to Consider**

- Volume of data (product catalog and price lists)
- Frequency of changes in business process impacting Pricing or Product Data
- Integrated systems and other data flows.
- Scalability and performance

## Considerations

### Product and Pricing Assumptions

The Connector lets you sync Simple Products with a flat price.

## Connector Guardrails

### Unsupported Product Types

To support the following B2B products, make other customizations in the Connector sync:

- Aggregate Products
- Assembly Products

### Unsupported Complex CPQ Product Bundles

Without other customizations, we don't recommend using complex CPQ Product Bundles with many Options and Constraints. They're not ideal for using with self-service or the Connector.

### B2B Product Entitlements

The Product and Pricing Sync process doesn't automate the B2B Product Entitlements. Associate The "Account Groups" to the "Price Lists" after completing the Product and Pricing Sync in B2B Commerce.

### Discounts and Promotions

We recommend that you manage [discounts and promotions for B2B Commerce](#) in B2B. Don't sync the information from CPQ.

## Data Strategy

### Connector Data Flow Strategy and Governance

Advanced product and pricing capabilities within CPQ aren't available in B2B. For example, Bundles and Rules.

### ERP as the Product and Pricing Master

Don't directly integrate B2B Commerce with the ERP. The functionality supported in the Connector and the approaches listed here assume that CPQ is the publishing data source for B2B Commerce.

### Product Index Synchronization

Synced Products in B2B appear only in a Storefront after refreshing the [Product Index](#).

## Related Content



Review earlier steps in this solution.

- [Workflow](#)
- [Design Considerations](#)



Take the next steps in this implementation.

- [Configurations](#)

# Configurations

---

Use these configurations to link cart to cash.

## Invoice and Payment Steps

Manage revenue with B2B Commerce and CPQ. This section contains definitions, followed by two strategies for adding business value to the user case and requirements.

### Definitions:

#### Payment Gateways

Gateways are third-party software that provides connectivity to various payment processing networks, such as Visa and American Express. The Payment Method object provides a token that the Payment Gateway can use to process a payment.

Use a Payment Gateway for customer security, including:

- Cybersource
- Authorize.net
- Payeezy
- Salesforce Billing, which integrates with any payment gateway or API integration

#### Payment Transactions

These records of attempted payments store information such as Invoice, Payment Method, Gateway used and response, and reason for failure (if applicable).

#### Payment Schedulers

The Payment Scheduler automatically collects electronic payments for outstanding invoices.

You can configure Payment Scheduler to run once or schedule recurring payment runs.

#### Self-Service Payments

They allow the customer to pay their own invoices.

You can expose a Payment Terminal through an Experience Cloud site or use Lightning Platform to create public websites and applications that directly integrate with your Salesforce organization.

## Strategy 1: Use Salesforce Billing Hosted Card Payment

Embed Salesforce Billing as an iFrame in the B2B Checkout flow for payment processing. Directly collect customer payments into Salesforce CPQ+. Salesforce Billing eliminates the need to integrate transactional data objects from the storefront data model to Salesforce.

Pros	Cons
<p>Salesforce Billing directly captures payment details (tokenized credit card or electronic payment details).</p> <p>PCI compliance is better because there's no transfer of secure information like credit card details.</p>	<p>The payment page has limited styling options.</p> <p>Authorization and Settlement (Charge) happen in a single action with the out-of-box payment gateway connector for billing.</p>

Salesforce Billing lets you manually collect and allocate payments or automate the payment process. You can then post the payment to keep your books up to date.

Apply and process CC and ACH payments through payment gateways and lockbox integrations. You can also apply manual payments.

#### Implementation Steps:

1. Enable the Billing HPP Component setting in Salesforce Billing.
2. Modify the component based on your org's payment and card-saving needs for the attributes.
3. There's no requirement for syncing payment details between B2B and Billing.
4. After the Payment Scheduler runs in Salesforce Billing, it processes the outstanding invoices with the selected payment method.
5. Payment Scheduler generates data for the "Payment" and "Payment Transaction" records.

## Strategy 2: Use Default Storefront Gateway and Custom Integration

Use the out-of-the box Payment Gateway with Storefronts and custom integration. Connect to Salesforce Billing or external ERP. Use out-of-the box Commerce Storefront to process payment directly in Commerce Storefront. It uses a default Payment Gateway Connector option.

Pros	Cons
<p>There are no changes to the out-of-box Payment Gateway connector or the checkout flow in B2B.</p> <p>Authorization and Settlement can happen as part of a separate transaction.</p>	<p>Storefront Gateway requires customization to sync the B2B Transaction Payment details to Billing.</p>

Process Payment using the default Payment Gateway connectors in B2B. Sync the Electronic Payment transaction details to Salesforce Billing.

#### Prerequisites

- Configure Payment Gateway in Salesforce Billing.
- Enable Hosted Payment Page (HPP) for the Org by contacting Salesforce support.
- Clone the Permission set "Hosted Payment User", and enable it.
- Enable "Allow Payment Using Card Hosted Payment Component" for the Experience site users.
- [Add Permissions](#) to the Community user profile.

#### Implementation Approach

1. To design a Component that loads the HPP component, [create the Lightning Component](#).
2. Publish the "ccrz\_\_E\_StoredPayment\_\_c" and "ccrz\_\_E\_TransactionPayment\_\_c" associated with the B2B Order to the Billing Payment Method "blng\_\_PaymentMethod\_\_c."
3. To publish Payment Method details to Billing, publish details from B2B to Salesforce Billing. [Refer to sample code](#). Use a trigger on the "CC Stored Payment" or "CC Transaction Payment" records. Note: the code snippet is for guidance only.
4. The lightning component has the "transactionType" as "SavePaymentCard." It ensures a capture of payment details without immediate processing.
5. Create a Lightning component with "force:cardPayment."
6. Create a Lightning app with the extension "ltng:outApp."

7. Create the Visual Force page for payments. [Refer to sample code.](#)
8. Update CC Admin Config:
  - a. Create a module from the CC Admin Storefront configuration setting, and select "Create Module."
  - b. The checkout module includes a newly created module.
  - c. Approve the Visual Force page in the Payment Module.

## Related Content



Review earlier steps in this solution.

- [Workflow](#)
- [Design Considerations](#)
- [Connector](#)