



---

# Offer Buy Online, Pick Up in Store Service (BOPIS)

Salesforce, Spring '21





# CONTENTS

- Offer Buy Online, Pick Up in Store (BOPIS) ..... 1
- Offer Buy Online, Pick Up in Store (BOPIS) Workflow ..... 2
- Design Considerations ..... 5
- Configurations ..... 7



# OFFER BUY ONLINE, PICK UP IN STORE (BOPIS)

Enable shoppers to buy products online for local store pickup through a Salesforce B2C Commerce storefront.



## Get Started



Explore system architecture related to this solution.

- [B2C Industry Blueprint](#)
- [B2C Reference Architecture](#)
- [B2C Solution Architectures](#)



Take Trailhead modules related to this solution.

- [Salesforce Solution Kits: Quick Look](#)
- [Customer 360 Guide for Retail: Quick Look](#)
- [Customer 360 Guides: Quick Look](#)

Give shoppers more options by expanding a storefront to offer stock pickup. Integrate Marketing Cloud for smooth post-purchase communications.

### **This solution kit helps your shoppers:**

- Reduce shipping costs.
- Receive merchandise faster.
- Take advantage of special discounts or promotions.
- Comply with social distancing preferences or requirements.

## Required Products

### Commerce Cloud

- Salesforce B2C Commerce Storefront Reference Architecture (SFRA) or SiteGenesis
- If using SFRA, "Plugin\_instorepickup" cartridge installed

### Marketing Cloud

- Journey Builder

- Content Builder Interactive Email
- Cloud Pages
- Optionally, Mobile Connect
- Optionally, Mobile Push

## Other System Requirements

---

- Salesforce Order Management, Enterprise Resource Planning (ERP), or another system that handles order management
- An inventory management system or the ability to send store inventory data via XML files to Salesforce Commerce Cloud

## Data Requirements

---

- Product Catalog, for any products you didn't already offer online and the BOPIS custom attribute
- Store Information Details, such as address, phone number, and hours
- If using Commerce Cloud Geolocation API, Store Geolocation Data to power the store locator
- Store Inventory Data, including stock keeping units (SKU) by store

## Recommended Skill Sets

---

- Marketing Cloud Email Specialist, preferably with Journey Builder experience
- Marketing Cloud Consultant
- Salesforce Commerce Cloud Certified Architect
- Salesforce B2C Commerce Cloud Certified Developer (front-end and back-end)
- Business owners from Commerce, Marketing, and Retail teams to define requirements

## Implement This Solution

---

### [Offer Buy Online, Pick Up in Store \(BOPIS\) Workflow](#)

Learn how data flows through the configurations when you let shoppers buy products online for local store pickup.

### [Design Considerations](#)

Keep these design considerations in mind when you let shoppers buy products online for local store pickup.

### [Configurations](#)

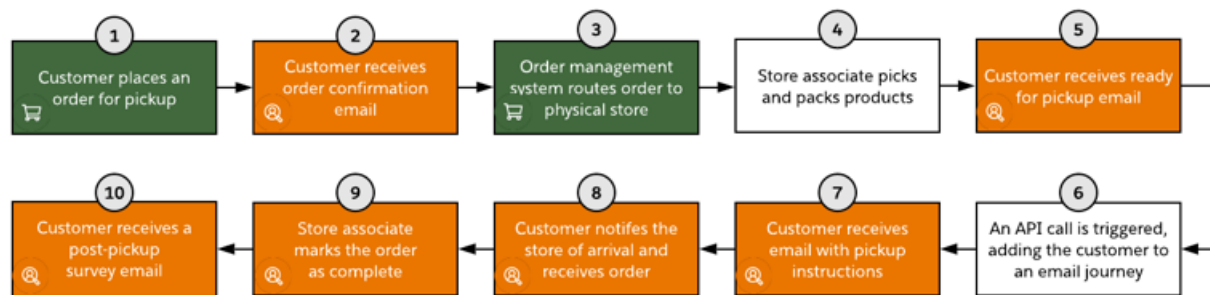
Use these configurations to let shoppers buy products online for local store pickup.

## Offer Buy Online, Pick Up in Store (BOPIS) Workflow

---

Learn how data flows through the configurations when you let shoppers buy products online for local store pickup.

## Workflow

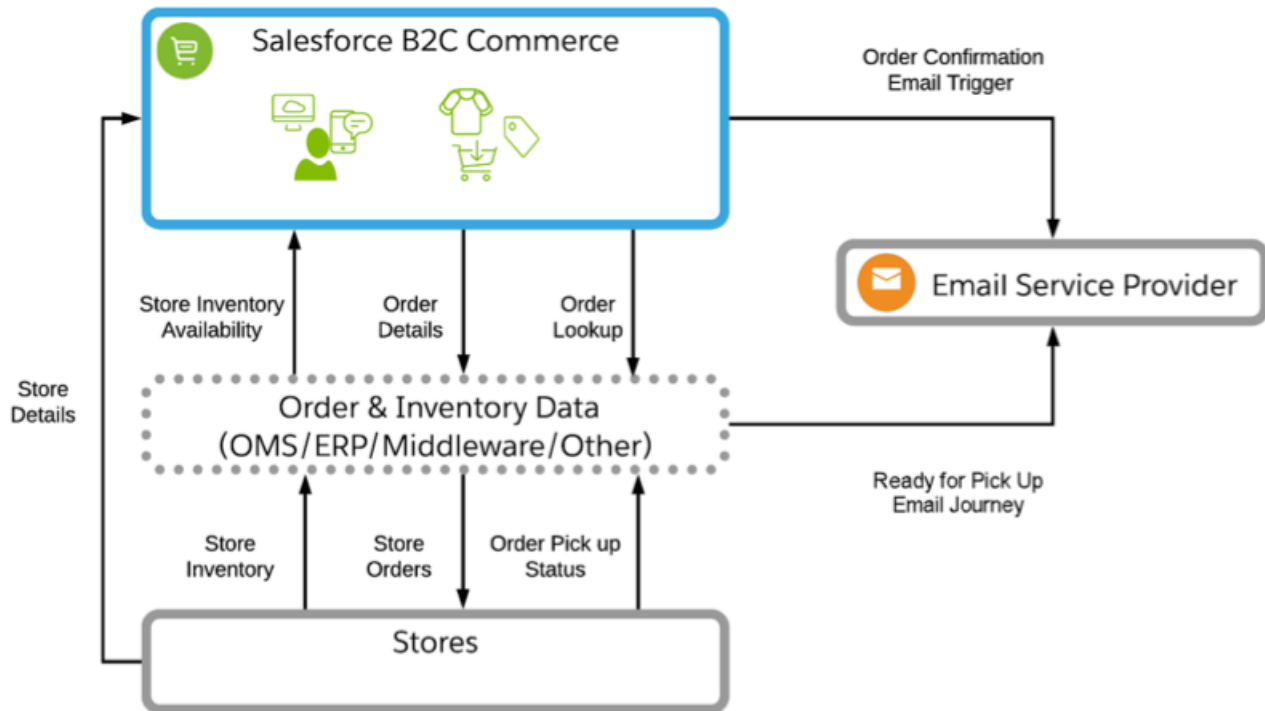


1. The shopper places an order on a B2C Commerce storefront, selecting store pickup from a local store that has available inventory. The order is exported to an order management system (OMS) or other internal system that processes and routes orders.
2. Marketing Cloud sends an order confirmation email to the customer.
3. The order management system routes pickup orders to the appropriate physical store.
4. A store associate reviews the order via an app or on the order management system console.
5. The associate picks and packs products to fill the order. The associate triggers an action to send an email notification to the customer that their order is ready for pickup with pickup instructions.
6. An API call adds the customer to an existing journey in Marketing Cloud.
7. The customer receives an interactive email that details pick up instructions.
8. Upon arrival at the store, the customer picks up their order using one of the methods provided. If the customer is conducting a contactless pickup, they click an "I have arrived" link in their email to notify the store of their arrival.
9. A store associate marks the order as complete.
10. Marketing Cloud sends a post-pickup survey to the customer.

## Example Data Flow



**Note:** This diagram represents one of many potential examples of systems and types of data flows that support BOPIS.



1. Collect the catalog, product, order, and customer [data feeds](#) from the B2C Commerce staging and production environments.
2. Move the data to the [Marketing Cloud SFTP server](#) so that Marketing Cloud consumes it on a schedule.
3. Use B2C Commerce Cloud data in Marketing Cloud via data extensions when authoring email templates, creating journeys, and developing personalized product recommendations.
4. Extend the B2C Commerce Cloud storefront by replacing its email functions with the Marketing Cloud Connector's trigger-send features. To trigger transactional emails for [common storefront scenarios](#), implement Marketing Cloud Connector customizations.
5. Implement the Marketing Cloud collect.js through the storefront by using the Connector's reference implementation. Predictive Intelligence handles abandonment scenarios targeting search, cart, and checkout. To implement behavioral monitoring, ensure that Predictive Intelligence is licensed as part of Marketing Cloud.
6. To drive behavioral messaging, personalization, and segmentation in Marketing Cloud, use the shopper behavioral data collected from collect.js.

## Functionality Considerations

This data flow configuration:

- Moves email authoring to Marketing Cloud's Email Studio.
- Enables transactional email delivery triggered by B2C Commerce Cloud and generated by Marketing Cloud.
- Enables email delivery via Marketing Cloud Journey Builder where B2C Commerce Cloud triggers the journey entry.
- Assumes that Marketing Cloud's interactive email form feature signals the customer's arrival for pickup.



## Related Content



Review this solution's use case and purpose.

- [Offer Buy Online, Pick Up in Store \(BOPIS\)](#) on page 1



Take the next steps in this implementation.

- [Design Considerations](#) on page 5
- [Configurations](#) on page 7

## Design Considerations

---

Keep these design considerations in mind when you let shoppers buy products online for local store pickup.

### Key Questions

#### Stores

- Which stores offer buy online, pick up in store (BOPIS)?
- How many stores offer BOPIS?
- Can each store support the space and logistics of holding packages for customer pickup?
- Must stores offer curbside pickup?
- What extra training do store associates, call center agents, and merchants need?

#### Products

- Which system holds product data for store items that aren't in your online catalog?
- How do stores select which products to offer for BOPIS?

#### Inventory

- What system holds your store inventory and how to access it?
- What type of threshold or safety stock do you account for, and how do you maintain it?

### Store Pick-Up Considerations

The process for routing orders to stores depends on your specific systems and implementation, including:

- An in-store app that store associates use to receive orders and mark them as ready for pickup
- In-store access to viewing and updating orders in your OMS or ERP system
- A more basic process for getting the order information to your stores for associates to pick and pack

Regardless of the method, develop a clear process for Store Associates to handle order pickup, customer inquiries, returns, and other interactions. Plan to notify customers that orders are ready for pickup. When picked up, close the loop on the process and mark orders fulfilled.

For more information, see [Communicate Store Changes](#) and [Schedule an Appointment](#).

## Curbside Pickup Considerations

- Place clear signage where customers wait.
- Provide a store phone number to call or text when they arrive.
- To offer contactless delivery, have associates put packages in their trunk or have a location outside the store where associates can place packages for pickup.

Have dedicated staff to communicate with customers and take packages to their car. For more information, see [Facilitate Curbside Pickup](#).

## Performance and Scalability Considerations

If using Service Cloud, consider using a common profile across Marketing, B2C Commerce, and Core. To learn more, review [Explore Cross-Cloud Engagement Data Models](#).

## Other OOTB Design Options

While not required, this section describes some of the more common storefront enhancements that merchants use.


Surface the Store Locator in the Header

1. This placement draws attention to the Store Locator. The shopper searches for a store and selects it before they browse to a product detail page.
2. Clicking the Store Locator link opens a modal window or directs the user to the Store Locator page.

 **Note:** Use an AJAX request and data injection to store the ID in a cookie and load the selected store details into the header.

To identify products available for store pickup, add a filter on your product listing page.

1. Make this filter display with other PLP refinements managed in Business Manager.
2. On the product tiles, use the custom attribute to trigger the addition of a callout message. This message can indicate that a product is available for BOPIS, such as “Free In-Store Pickup.”

 **Note:** This customization uses the `availableForInStorePickup` attribute on products. It determines whether a flag or callout message shows on a product tile to denote that an item is sold in stores. It’s not a check for store inventory availability. That check happens only on the product detail page with a selected SKU.


Add a form field entry in Checkout for the shopper to specify an extra person to pick up the order.

 **Note:** Update your Order Export file and adding the additional data attributes.

## Marketing Cloud Key Considerations

Consent Management

Honor and respect customer wishes when they request specific forms of contact from your company or opt out of certain types of data sharing. We provide [Consent Management Details](#) to help you determine the best way to comply with the data protection and privacy regulations that apply to your company.

 **Note:** This solution kit refers to sending transactional emails. To send promotional emails to the same subscriber, the subscriber must explicitly opt in to receive marketing communications. Similarly for Mobile Connect, the subscriber must explicitly opt in to receive marketing communications. MobilePush users must obtain consent for contacts to receive push messages.

## Marketing Cloud Contact Model (Omni-Channel Messaging)

### Contact Key

A single Contact Key manages and relates the [contact](#) through the different channels. The Contact Key is a unique identifier that you assign to a contact. If a subscriber gets an email and wants to be on mobile, the Contact Key adds the contact to the mobile channel. The Contact Key identifies a contact within an account and ties together the contact, channels, and the relationship. The Contact Key is the same no matter what channel is used to send messages.

The Contact Key allows you to connect contacts in multiple channels. Without the Contact Key, it would be difficult for Marketing Cloud to know to connect the contact because the contact has two different identifiers. Marketing Cloud processes the information as two different contacts in Contact Builder. Make sure that you're consistent across all channels when assigning a Contact Key to a contact.

### Mobile Push

Set the [Subscriber Key](#) or [Contact Key](#) to the correct key in MobilePush.

There are several ways of setting the Contact Key on the MobilePush channel to be the Salesforce ID. One of the most common methods is to set up a backend service that syncs Salesforce IDs to an external system. When a mobile user identifies themselves in the app, the service figures out who they are and what their Salesforce ID is. The service returns that to the mobile app and sets it as the Contact Key.

### Mobile Connect

When using the [QueueMO opt-in method](#), the client specifies the Subscriber Key.

## Related Content



Review earlier steps in this solution.

- [Offer Buy Online, Pick Up in Store \(BOPIS\) Workflow](#) on page 2



Take the next steps in this implementation.

- [Configurations](#) on page 7

## See Also

- [Communicate Store Changes](#)
- [Schedule an Appointment](#)
- [Facilitate Curbside Pickup](#)

## Configurations

---

Use these configurations to let shoppers buy products online for local store pickup.

The following configuration is for buy online, pick up in store (BOPIS) using B2C Commerce Cloud out-of-the-box (OOTB) functionality and storefront customizations. The steps here are for storefronts built with Storefront Reference Architecture (SFRA).

We also provide links to other supporting documentation on the B2C Commerce Cloud Documentation Portal and the B2C Commerce Cloud GitHub repository. For SiteGenesis, refer to the documentation on the infocenter for [SiteGenesis in-Store Pickup](#), as there are some minor variations.

## Install the Plugin and Supporting Metadata

Follow the [instructions for getting started](#). Ensure that your instance has the proper system object extensions metadata to support the plugin, including the following object custom attributes:

- **Product: availableForInStorePickup (boolean)**  
When set to true, this attribute enables functionality to find a product in a store via the product detail page. It also determines when an InventoryList (or external API inventory ID) for a given store must have a ProductLineItem when added to the Basket.
- **ShippingMethod: storePickupEnabled (boolean)** determines if the shipping method shows as an in-store shipping method.
- **Store: inventoryListId (string)** stores an ID reference to a given store's associated inventory list (or external inventory ID when using an API). This inventory list determines product availability. Upon add to cart, associates with the ProductLineItem via the productInventoryListID attribute.
- **ProductLineItem: fromStoreId (string)** stores an ID reference to a given Store object on a line item level.
- **Shipment: fromStoreId (string)** stores an ID reference to a given Store object on a shipment level.
- **Shipment: shipmentType (string)**  
For shipments where this attribute is set to "instore," only shipping methods where storePickupEnabled = true are available.

## Data Setup and Considerations

### Products

Update your existing product feed to account for the availableForInStorePickup custom attribute.

### Inventory

Choose different ways to retrieve store inventory data to power BOPIS functionality on your storefront.

- Regularly extract inventory data from your source system, and send it to B2C Commerce Cloud as XML formatted Inventory Lists. For the XML file format, refer to the [inventory.xsd schema](#).
- Use APIs to call your source system when you must retrieve store inventory.

### Stores

[Add your store data](#) in Business Manager, or import store information in an XML file. If importing via an xml file, follow the [stores.xsd schema](#).

Another option for implementing the ability for a customer to shop a selected store is to use price books within B2C Enterprise. The steps to implement this code on your site are in the [Shop By Store Github](#).

### Geolocation

If using B2C Commerce Cloud API for Geolocation data, import all [Store Geolocation data](#) following the [geolocation.xsd schema](#).

Use the following job steps to perform automated maintenance of your data:

- [ImportInventoryLists](#)
- [ImportStores](#), if importing stores via XML
- [ImportCatalog](#), to import updates to availableForInStorePickup attribute on products

## Resolving Template Conflicts

With multiple plugins, you can resolve multiple template conflicts. Salesforce provides another cartridge, [plugin\\_cartridge\\_merge](#), which provides a reference for how to merge together multiple plugins in a single customization cartridge. Refer to this cartridge for inspiration when considering the best way to implement plugins in your existing code base.

- Review the list of [templates that override the base SFRA cartridge](#).
- Compare template code between `app_storefront_base` (main SFRA cartridge) and `plugin_instorepickup` (main BOPIS plugin cartridge). Use a comparison tool to compare the files for differences, or run a “diff” of the files.
- Copy the identified delta code added from the `plugin_instorepickup` cartridge to any customizations cartridge code that overrides it. If the templates being overridden don't exist in any cartridges other than `app_storefront_base`, don't copy those changes to any other cartridges. Instead, use the `plugin_instorepickup` cartridge's version of those templates.

To view the expected cartridge stack, see [SFRA Overview](#).

- To ensure that plugin features work correctly, test the storefront functionality, and review all logs for errors.

## Storefront Template Customizations

With the in-store pickup cartridge installed, the Storefront Reference Architecture provides key out-of-the-box functionality and template code to support BOPIS. See the [SFRA Wireframes Documentation](#). If using Site Genesis as your reference application, see the OOTB design and functionality for BOPIS in the [Site Genesis Wireframes and Functional Spec Documentation](#).


The following are the key areas to update when enabling the OOTB BOPIS functionality. They depend on your current template design.

- Product Detail Page
- Cart
- Checkout
- Order Confirmation / Order History & Details

Other customized sections that further enhance the customer experience include:

- Header
- Product Listing / Search Results Page

To identify the customized areas, compare your current pages with the OOTB template designs. Look for areas that include elements to support BOPIS. This guide covers the minimum OOTB features to support BOPIS. Work with your technical teams and design teams to help identify areas where your storefront templates need adjustments. Support the functionality, and adhere to your brand's look and feel.

 **Note:** It's important to use clear and prominent messaging about the store pick-up process and FAQs.

As with any enhancements you make to your storefront, follow your organization's process for thoroughly testing. Ensure that your data is accurate. Test data flows across all systems. Make sure that impacted teams know about any changes to their process. Before launching your new BOPIS offering, ensure that all storefront functionality works as expected.

## Marketing Cloud Configuration Steps

Trigger an Order Confirmation Email in the Marketing Cloud with the Marketing Cloud [Transactional Email Solution Kit](#).

Create an “Order is ready for pickup journey” that uses an [API event](#) as its entry source. Inject the subscriber into the Marketing Cloud “Order is ready for pickup” journey when the order fulfills. Marketing Cloud's REST API offers you access to the Journey Builder API. You inject subscribers into a journey via an API call. Configure this REST service call to Marketing Cloud \using a B2C Commerce Cloud Service. The call uses the [fire entry event](#) method to inject a subscriber into the journey.

 **Note:** This call triggers from the order fulfillment page after the order processes in B2C Commerce Cloud or OMS.

Include the following required elements to make this journey an Omni-Channel journey. They send messaging that the order is ready to be picked up at the store. If your only channel is email, skip the mobile steps. If using only mobile, skip the email steps.

1. SMS Order is ready for pickup message
2. MobilePush Order is ready for pickup message
3. An [interactive email](#) with a button or a similar call to action that the consumer clicks to alert the service agent that they arrived. Add other identifying information like make and model of the car and parking location to this email.
4. An [interactive landing page](#) when the consumer clicks the button that:
  - Updates the OMS to indicate that the order is ready for pickup
  - [Sends an SMS](#) to a designated contact for the store using REST API
  - [Sends an email](#) to an alias at the fulfillment desk at the store
  - Updates the client system via API
5. A post-purchase survey email

## Related Content



Review earlier steps in this solution.

- [Offer Buy Online, Pick Up in Store \(BOPIS\) Workflow](#) on page 2
- [Design Considerations](#) on page 5

## See Also

- Get answers to your questions from B2C Commerce experts and other customers via the [B2C Commerce Trailblazer Community Group](#).
- Search and find specifications and technical info for all aspects of B2C Commerce on the [Salesforce B2C Commerce Infocenter](#).
- Review the [Calendar](#) of upcoming Commerce Cloud learning opportunities.
- [Salesforce Order Management](#)
- Request a [Commerce Cloud Accelerator](#).
- If you need expert guidance for planning your BOPIS solution, contact your Customer Success Manager to ask about Advisory Services.