



Salesforce B2C Commerce Implementation Playbook:

Buy Online Pick up In Store (BOPIS)

Last updated: April 2020



About this Playbook	3
Benefits of offering BOPIS	3
Use Case Overview	4
High Level Sample BOPIS Workflow	4
Implementation Prerequisites	5
Required Products	5
Required Systems or Services	5
Data Requirements	6
Expertise Needed	6
Example Data Flow	6
Implementation Steps	7
Install Plugin & Supporting Metadata	7
Data Setup & Considerations	8
Resolving Template Conflicts	8
Storefront Template Customizations	9
Test your solution!	24
Optional Additional Enhancements	24
Store Pick-up Considerations	26
Additional Resources	27



About this Playbook

The purpose of this document is to help merchants and developers in customizing their existing Salesforce B2C Commerce storefront, to enable functionality that gives a shopper the option to buy a product online, and pick it up at a local store.

This playbook uses standard Commerce Cloud functionality, as well as suggestions for how to make customizations to meet each merchant's specific business needs and to follow industry best practices. Depending on how your storefront has been implemented, there may be more or less customization required to enable this functionality.

And while this Playbook will provide some guidance on functionality that will be required on downstream platforms and systems, it's primary focus is on the aspects related to storefront customizations on Commerce Cloud.

Before you begin any project that involves customizing your storefront, make sure you align with your business and technical teams, and any other teams that may be impacted. If you have a Certified Partner you're already engaged with, involve them in your plans.

Salesforce experts are here to help! If you need support implementing this functionality, our Commerce Cloud Professional Services teams can help get you up and running. From Certified Technical Architect guidance with your internal or Partner development teams, or hands-on Certified Commerce Cloud Developers to implement the solution for you. Reach out to your Commerce Cloud Customer Success Manager to start a conversation.

Benefits of offering BOPIS

Enhancing your storefront to offer BOPIS has many benefits for your shoppers and you as a merchant.

Many shoppers like the convenience of BOPIS in order to:

- avoid paying shipping costs
- get merchandise faster; enjoy instant gratification
- take advantage of special discounts or promotional offers

And merchants will like the impact on their bottom line:

- The greater majority of shoppers who go into a store to pick up an online order, end up making additional purchases.
- The more orders that customers pick up themselves at a store, the more money that is saved in shipping carrier costs, and other fulfillment logistics expenses.

Use Case Overview

BOPIS, BOPUS, In-Store Pick Up, Click and Collect, Curbside Pick up! Are these the same things? Different? And what do I need to do to offer this to my shoppers?

No matter what you call it, the ability for a shopper to purchase a product on a brand's ecommerce storefront, and be able to pick it up at a local brick and mortar store has become table stakes functionality for many retailers. And whether you have a dozen stores in one state or hundreds spread out across countries, there are some aspects that are a part of any complete BOPIS solution.

Let's look at a high level BOPIS journey, identifying 5 key actions.

High Level Sample BOPIS Workflow



Note: This is a high-level view of what a typical process could look like. The solution you plan may involve more or less steps, or involve other systems than are represented here.

1. First, the shopper places the order on your Commerce Cloud storefront indicating that they would like to pick up the item from a store that has available inventory, and the order is exported down to an Order Management System (OMS).
 - a. Don't have an OMS? You may have an ERP, DOM, or some other internal system that processes and routes your orders. For the purposes of this flow, let's just call that your OMS.
 - i. [Click here to learn about our Salesforce Order Management solution](#), which integrates directly into B2C Commerce and natively with Service Cloud.
2. Next, the OMS identifies orders that are to be fulfilled from a physical store, and routes those orders to the appropriate store.

3. At the store, whether via an in-store app, access to an OMS console, or some other process, the store associate will review the order, pick and pack the products on the order, and trigger an action to send an email or other notification to the customer that their order is ready for pick up.
4. When the customer arrives at the store, they may go inside to a specific location in the store, ask a sales associate, or even just pull up in their car “curbside” for an associate to give them their order “contactless”.
5. Lastly, with the customer having received their order, the order cycle is complete, and the order is marked as picked up, thereby completing the flow.

Implementation Prerequisites

For this guide, we’ll be looking at the implementation aspects for Commerce Cloud, focused on the storefront part of the BOPIS workflow, and what is required for this functionality. Depending on your system architecture and how you’ve already implemented your site, your BOPIS implementation may include additional requirements.

First, here are some questions you’ll need to answer to plan your approach:

- Stores
 - What stores do you want to offer BOPIS at? One to start? Only some stores? All stores?
 - Can each store support the space and logistics of holding packages for customer pick-up?
 - Need to offer Curbside pick-up? What will your process be for handling this?
 - What additional training will your Store Associates, Call Center Agents, and Merchants need?
- Products
 - What system holds your product data for store items that are not currently part of your online catalog?
 - Do you have a way to identify which store products you will offer for BOPIS?
- Inventory
 - What system holds your store inventory and how will you access it?
 - What type of “threshold” or “safety stock” will you need to account for and how and where will you maintain this?

Now, let’s take a look at the prerequisites for the Commerce Cloud storefront functionality.

Required Products

- Salesforce B2C Commerce (Storefront Reference Architecture “SFRA” or SiteGenesis)
- “plugin_instorepickup” cartridge installed (if using SFRA)

Required Systems or Services

- An Order Management System, ERP, or another system that handles order management

- An inventory management system/service, or the ability to send store inventory data via XML file to Salesforce Commerce Cloud at regular, frequent intervals, or via real-time API.
- Salesforce Marketing Cloud, another email service provider, or system that you currently use to send transactional emails to your customers.

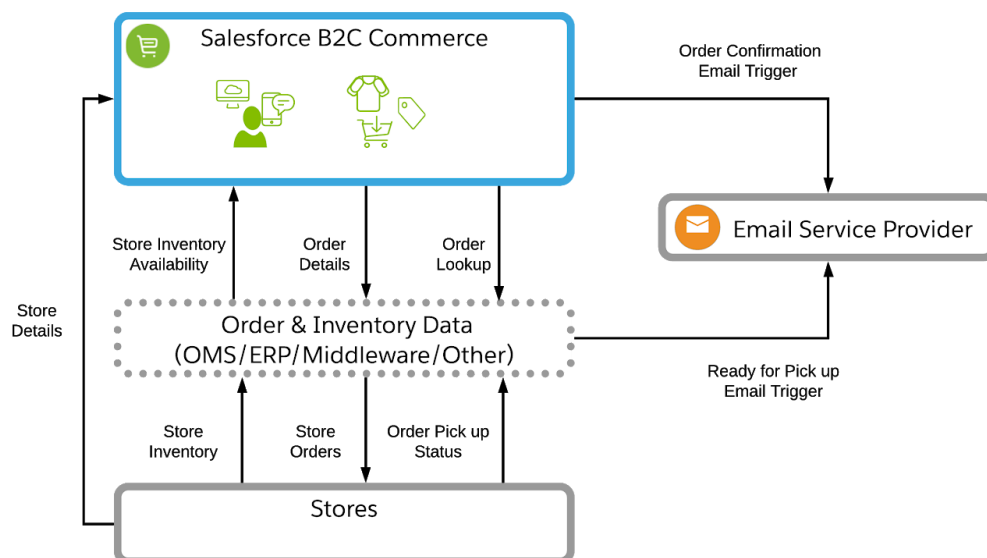
Data Requirements

- Product Catalog (for any products you did not already offer online, and BOPIS custom attribute)
- Store Information Details (such as address, phone number, hours)
- Store Geolocation Data (to power the store locator if using Commerce Cloud Geolocation API)
- Store Inventory Data (sku-level inventory by store)

Expertise Needed

- Salesforce Commerce Cloud Certified Architect
- Salesforce Commerce Cloud Certified Developer(s) (front-end and back-end)

Example Data Flow



Note: This diagram represents one example of the systems that could be involved, and the types of data flows you could have to support BOPIS. Your solution may involve different systems or data flows.



Implementation Steps

We've documented here the main steps for implementing BOPIS, using Commerce Cloud OOTB functionality and storefront customizations. There are links out to other supporting documentation on the **Commerce Cloud Documentation Portal** and the **Commerce Cloud GitHub repository**. The steps here are for storefronts built with SFRA. For SiteGenesis, please refer to the documentation on the infocenter for [SiteGenesis in-Store Pickup](#), as there are some minor variations.

1. Install Plugin & Supporting Metadata

Follow instructions in Getting Started section here:

https://github.com/SalesforceCommerceCloud/plugin_instorepickup#getting-started

Ensure your instance has access to the proper system object extensions metadata to support the plugin:

https://github.com/SalesforceCommerceCloud/storefrontdata/blob/master/demo_data_sfra/meta/system-objecttype-extensions.xml

This includes the following object custom attributes:

- **Product: availableForInStorePickup (boolean)**
This attribute, when set to **true**, is used to enable the functionality to find a product in a store via the PDP. It is also then used to determine when an InventoryList (or external inventory ID when using an API) for a given store needs to be associated with a ProductLineItem when it is added to the Basket.
- **ShippingMethod: storePickupEnabled (boolean)**
Determines if the shipping method appears as an in-store shipping method.
- **Store: inventoryListId (string)**
This attribute is used to store an ID reference to a given store's associated inventory list (or external inventory ID when using an API). This will be the inventory list used to determine product availability. Upon add to cart, it will also be associated with the ProductLineItem via the **productInventoryListID** attribute.
- **ProductLineItem: fromStoreId (string)**
This attribute is used to store an ID reference to a given Store object on a line item level.
- **Shipment: fromStoreId (string)**
This attribute is used to store 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 will be available.

2. Data Setup & Considerations

Products:

You will want to update your existing product feed to account for the **availableForInStorePickup** custom attribute.

Inventory:

There are different ways you can choose to retrieve store inventory data to power BOPIS functionality on your storefront.

- Extract inventory data from your source system and send it to Commerce Cloud as XML formatted Inventory Lists, updating at regular intervals
 - For the XML file format, refer to the [inventory.xsd schema](#). ([schema](#)).
- Use APIs to call your source system when you need to retrieve store inventory

Either approach will work, although the approach you take will determine certain aspects of the implementation. If using an API-based approach, wherever Commerce Cloud would look for a store inventory list, you will now need to use an API to call the appropriate store's inventory details. Keep that in mind as you plan your solution.

Stores:

You can [add your store data manually](#) in Business Manager, or import store information in an XML file.

- If importing via an xml file, follow the stores.xsd [schema](#).

Geolocation:

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

You may need to leverage the following job steps as needed to perform automated maintenance of your data:

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

3. Resolving Template Conflicts

You may be using multiple plugins, and therefore you may have multiple template conflicts to resolve. Salesforce provides another cartridge, [plugin_cartridge_merge](#) which provides a reference for how to

merge together multiple plugins in a single customization cartridge. You can refer to this cartridge for inspiration when considering the best way to implement plugins in your existing code base.

- Review the following list of templates which override the base SFRA cartridge:
https://github.com/SalesforceCommerceCloud/plugin_instorepickup#template-conflicts
- Compare template code between app_storefront_base (main SFRA cartridge) and plugin_instorepickup (main BOPIS plugin cartridge). This involves using a comparison tool to compare the files for differences, or run a “diff” of the files as shown in the following example:

<pre> <!-- Availability --> <div class="row availability product-availabi <isinclude template="product/components/a </div> </pre>	<pre> 1 1 >> 2 2 << 3 3 4 4 >> 5 5 6 << 7 8 9 10 11 12 </pre>	<pre> <!-- Availability --> <div class="availability row product-availabi <isinclude template="product/components/a </div> <isif condition="\$\${!isQuickView}"> <!-- In store inventory --> <div class="pdp-in-store-inventory"> <isinclude template="product/componen </div> </isif> </pre>
---	---	--

Example of code comparison for productAvailability.isml.

During comparison, changes are highlighted to show where the differences are. In the above example, app_storefront_base code is on the left and the plugin_instorepickup code is on the right. New code is added to lines 6-12 by the version of the ISML template used by the plugin.

- Copy the identified delta code added from the plugin_instorepickup cartridge to any customizations cartridge code which may override it. If the templates being overridden by the plugin_instorepickup cartridge don't exist in any cartridges other than app_storefront_base, you should not need to copy those changes to any other cartridges and instead use the plugin_instorepickup cartridge's version of those templates.

For example, if productAvailability.isml is in your custom cartridge, since custom cartridges override base and plugin cartridges alike, the code delta mentioned above from lines 6-12 would need to be added to your custom cartridge override. (see [SFRA Overview](#) to see the expected cartridge stack)

- Test the storefront functionality and review all logs for errors to ensure all of the plugin features work correctly.

4. Storefront Template Customizations

The Storefront Reference Architecture (SFRA), with the in-store pick up cartridge installed, provides key out of the box (OOTB) functionality and template code, to support BOPIS. The below screenshots and



functional details are from the [SFRA Wireframes Documentation](#). If you are using Site Genesis as your reference application, you can see the OOTB design and functionality for BOPIS in the [Site Genesis Wireframes and Functional Spec Documentation](#).

For enabling the OOTB BOPIS functionality, here are the key areas that you will need to update, depending on your current template design.

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

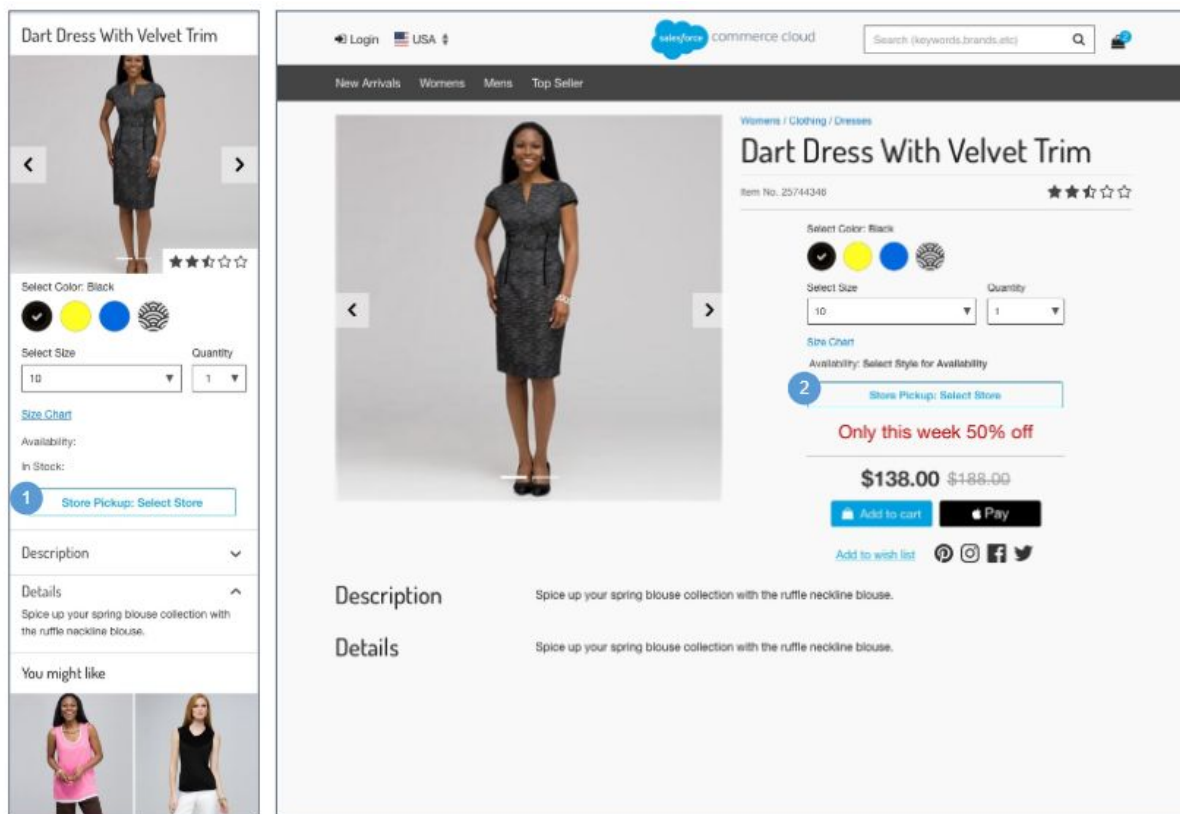
We've also included some additional sections you may choose to customize, to further enhance the customer experience, or provide additional custom functionality. These sections include:

- Header
- Product Listing / Search Results Page

To identify the areas that you will need to customize, based on your specific storefront implementation, you can compare your current pages with the OOTB template designs for areas that include elements to support BOPIS. While this guide covers the minimum OOTB features to support BOPIS, you may choose to make additional changes in support of your business requirements or brand style guide. Work with your technical teams and design teams, to help identify areas where your storefront templates may need more or fewer adjustments to support the functionality and adhere to your brand's look and feel.

Note: Don't forget about the importance of clear and prominent messaging throughout your site experience about the store pick-up process and FAQs!

Product Detail Page (no saved store) - OOTB Example Design:



Product Detail Page (no saved store) - OOTB Functional Details:

The Product Detail page for a product with in-store pickup has the same functionality as the Product Detail page for a standard product or a variation product, with the following exceptions

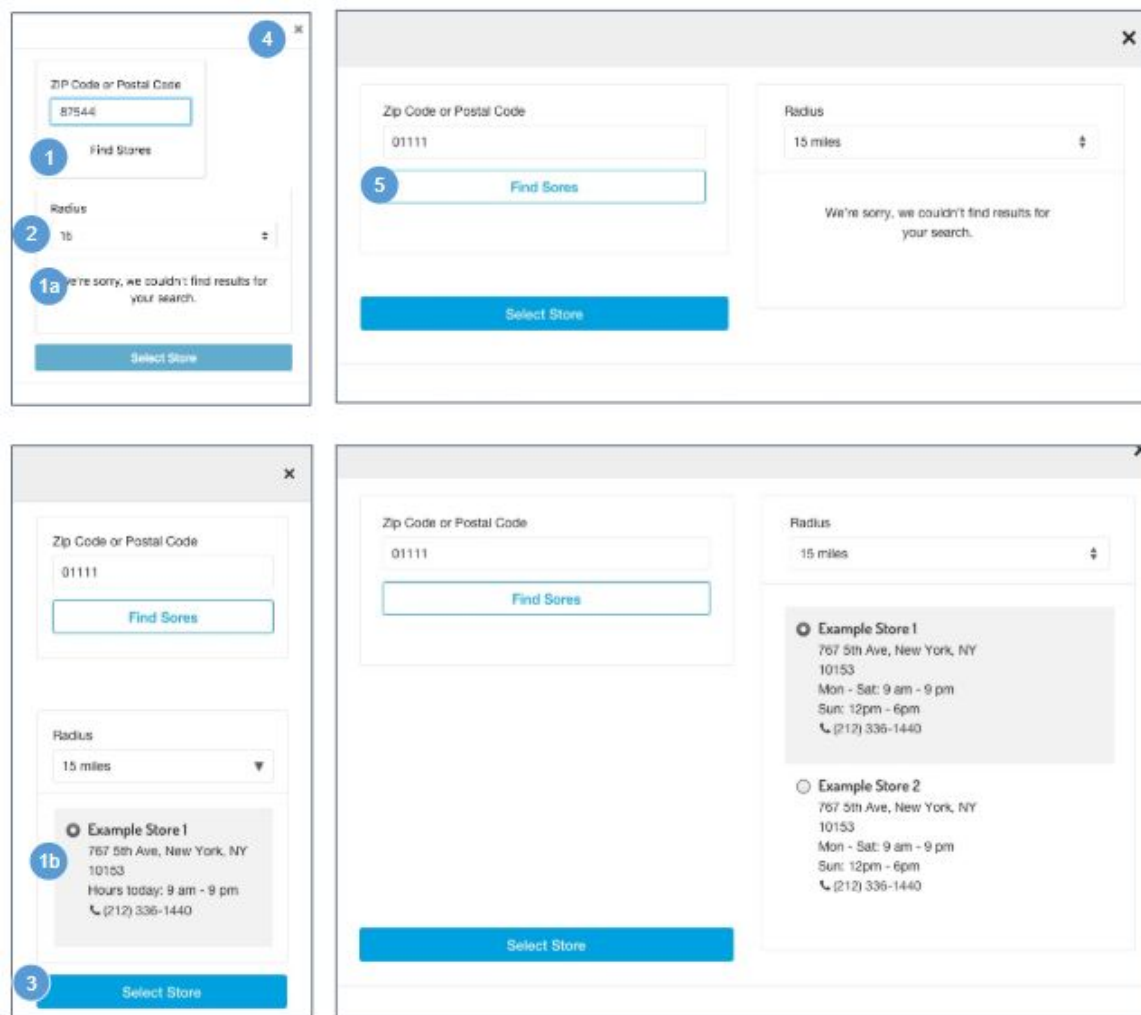
Mobile:

1. Clicking Store Pickup: Select Store opens the Store Selector pane.

Desktop:

2. Clicking Store Pickup: Select Store opens the Store Selector page in a modal window.

Store Selector (no saved store) - OOTB Example Design:



The screenshots illustrate the Store Selector interface in four states:

- Top Left (State 1):** Initial search state. The 'Zip Code or Postal Code' field contains '87544'. The 'Radius' is set to '10'. A message below the radius field states: 'We're sorry, we couldn't find results for your search.' A 'Find Stores' button is visible.
- Top Right (State 2):** Search state with '01111' in the zip code field and '15 miles' in the radius field. A 'Find Stores' button is highlighted.
- Bottom Left (State 3):** Search state with '01111' in the zip code field and '15 miles' in the radius field. A 'Find Stores' button is highlighted.
- Bottom Right (State 4):** Results state showing two example stores:
 - Example Store 1:** 767 5th Ave, New York, NY 10153. Mon - Sat: 9 am - 9 pm. Sun: 12pm - 6pm. ☎ (212) 336-1440.
 - Example Store 2:** 767 5th Ave, New York, NY 10153. Mon - Sat: 9 am - 9 pm. Sun: 12pm - 6pm. ☎ (212) 336-1440.

Store Selector (no saved store) - OOTB Functional Details:

Mobile:

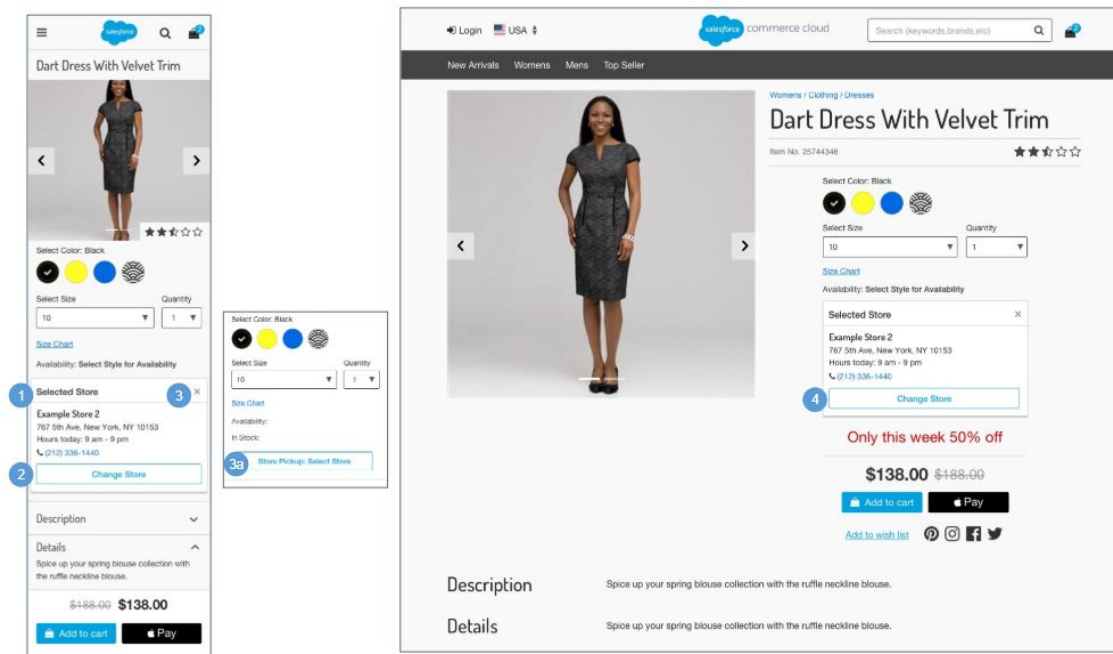
1. The shopper can enter a zip code or a postal code. Clicking Find Stores searches the geographical area specified.
 - a. Shows an error message if an incorrect postal code or zip code is entered, or if there are no matching stores for the entered code.
 - b. The system returns all valid store locations in list view based on the zip code / postal code and radius. The following store attributes are shown:
 - i. Store Name
 - ii. Address - Clicking opens Google Maps in a browser window with the store location pinned on the map. Store attributes for latitude and longitude must be configured in Business Manager.

- iii. Store Hours
 - iv. Phone Number - Clicking opens a prompt to dial the store's number.
2. Radius dropdown values are stored in the plugin code (15-300 miles), the default is 15 miles. Selecting a radius value refines the search results.
 - a. Miles can be converted to kilometers via the script method `'dw.catalog.StoreMgr.searchStoresBy.'`
 3. Clicking Select Store sets the store as the preferred store for the remainder of the session. The Store Selector panel closes, and the shopper is returned to the prior page showing the store information.
 4. Clicking Close (X) closes the panel and returns the shopper to the Product Detail page, which shows no store information.

Desktop:

5. Clicking Find Stores opens the Store Selector in a modal window (with or without results).

Store Selector (with saved store) - OOTB Example Design:



Store Selector (no saved store) - OOTB Functional Details:

The Product Detail page for a product with in-store pickup has the same functionality as the Product Detail page for a standard product or a variation product, with the following exceptions

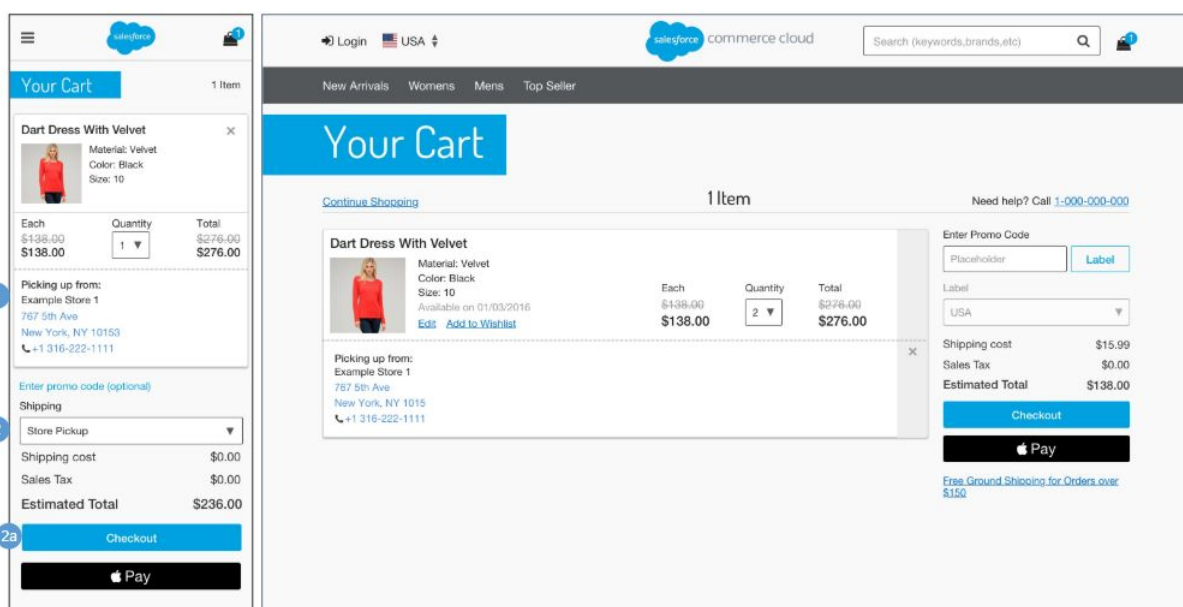
Mobile:

1. Shows the following store attributes:
 - a. Store Name
 - b. Store Address - Clicking opens Google Maps in a browser window with the store location pinned on the map. Store attributes for latitude and longitude must be configured in Business Manager.
 - c. Store Hours
 - d. Phone Number - Clicking opens a prompt to dial the store's number.
2. Clicking Change Store opens the Store Selector panel.
3. Clicking X removes the saved store. The Change Store button is replaced with Store Pick-up: Select Store.
 - a. Clicking Store Pickup: Select Stores opens the Store Selector panel.

Desktop:

4. Clicking Store Pickup: Select Store opens the Store Selector page in a modal window.

Shopping Cart / Basket - OOTB Example Design:





Shopping Cart / Basket - OOTB Functional Details:

The cart page for a product with in-store pickup has the same functionality as the cart page for a standard product or a variation product with the following exceptions.

Mobile / Desktop

1. If the shopper has a store saved in their session, shows the text 'Picking up From' and the following store attributes:
 - a. Store Name
 - b. Store Address - Clicking opens Google Maps in a browser window with the store location pinned on the map. Store attributes for latitude and longitude must be configured in Business Manager.
 - c. Phone Number - Clicking opens the phones prompt to dial the store's number.
2. If the shopper does not have a store saved in their session, they can select Store Pickup in the Shipping Method dropdown.
 - a. For guest shoppers, clicking Checkout opens the Checkout Login page.
 - b. For registered shoppers, clicking Checkout opens the Shipping Address | Shipping Method page with the Store Selector panel blank.

Shipping Address | Shipping Method (No Saved Store) - OOTB Example Design:

Checkout

2 Items \$138.00

Shipping

☐ Ship to or pick up from multiple locations?

Shipping Method:

Ground (arrives 04-18-2016) \$15.99

Next-Day (arrives 04-18-2016) \$20.99

Overnight (arrives 04-18-2016) \$29.99

Store Pickup

1 Zip Code or Postal Code

02111

2 Find Store

3 Radius

15 miles

1a We're sorry, we couldn't find results for your search.

4 Select Store

Payment

Order Summary

Subtotal \$138.00

Shipping \$15.99

Sales Tax \$0.00

Total \$153.99

2 Items \$476.00

Dart Dress With Velvet

Material: Velvet Color: Black Size: 10 Available in 01/03/2016

Each \$138.00 Quantity 2 Total \$276.00

Dart Dress With Velvet

Material: Velvet Color: Black Size: 10 Available in 01/03/2016

Each \$136.00 Quantity 1 Total \$136.00

5 Next Payment

Checkout

2 Items \$138.00

Shipping

☐ Ship to or pick up from multiple locations?

Shipping Method:

Ground (arrives 04-18-2016) \$15.99

Next-Day (arrives 04-18-2016) \$20.99

Overnight (arrives 04-18-2016) \$29.99

Store Pickup

1b Zip Code or Postal Code

02111

Find Store

Radius

15 miles

Example Store 1

767 1st Ave, New York, NY 10013

Mon - Sat: 9 am - 8 pm

Sun: 12 pm - 4 pm

☎ (212) 336-1418

Example Store 2

767 1st Ave, New York, NY 10013

Mon - Sat: 9 am - 8 pm

Sun: 12 pm - 4 pm

☎ (212) 336-1418

4a Select Store

Payment

Order Summary

Subtotal \$138.00

Shipping \$15.99

Sales Tax \$0.00

Total \$153.99

2 Items \$476.00

Dart Dress With Velvet

Material: Velvet Color: Black Size: 10 Available in 01/03/2016

Each \$138.00 Quantity 2 Total \$276.00

Dart Dress With Velvet

Material: Velvet Color: Black Size: 10 Available in 01/03/2016

Each \$136.00 Quantity 1 Total \$136.00

4a Next Payment

Checkout

Next Step? Call 1-800-XXX-XXXX

Shipping

☐ Ship to or pick up from multiple locations?

Shipping Method:

Ground (arrives 04-18-2016) \$15.99

Next-Day (arrives 04-18-2016) \$20.99

Overnight (arrives 04-18-2016) \$29.99

Store Pickup

Zip Code or Postal Code

Find Store

Radius

15 miles

We're sorry, we couldn't find results for your search.

Select Store

Order Summary

Subtotal \$138.00

Shipping \$15.99

Sales Tax \$0.00

Total \$153.99

2 Items \$476.00

Dart Dress With Velvet

Material: Velvet Color: Black Size: 10 Available in 01/03/2016

Each \$138.00 Quantity 2 Total \$276.00

Dart Dress With Velvet

Material: Velvet Color: Black Size: 10 Available in 01/03/2016

Each \$136.00 Quantity 1 Total \$136.00

Payment

Next Payment

5a

Before you can continue to the next step, you must select a store.

Shipping Address | Shipping Method (No Saved Store) - OOTB Functional Details:

The shopper is presented this page if they do not have a saved store in their session. The shopper can select Store Pickup from the Shipping Method panel.

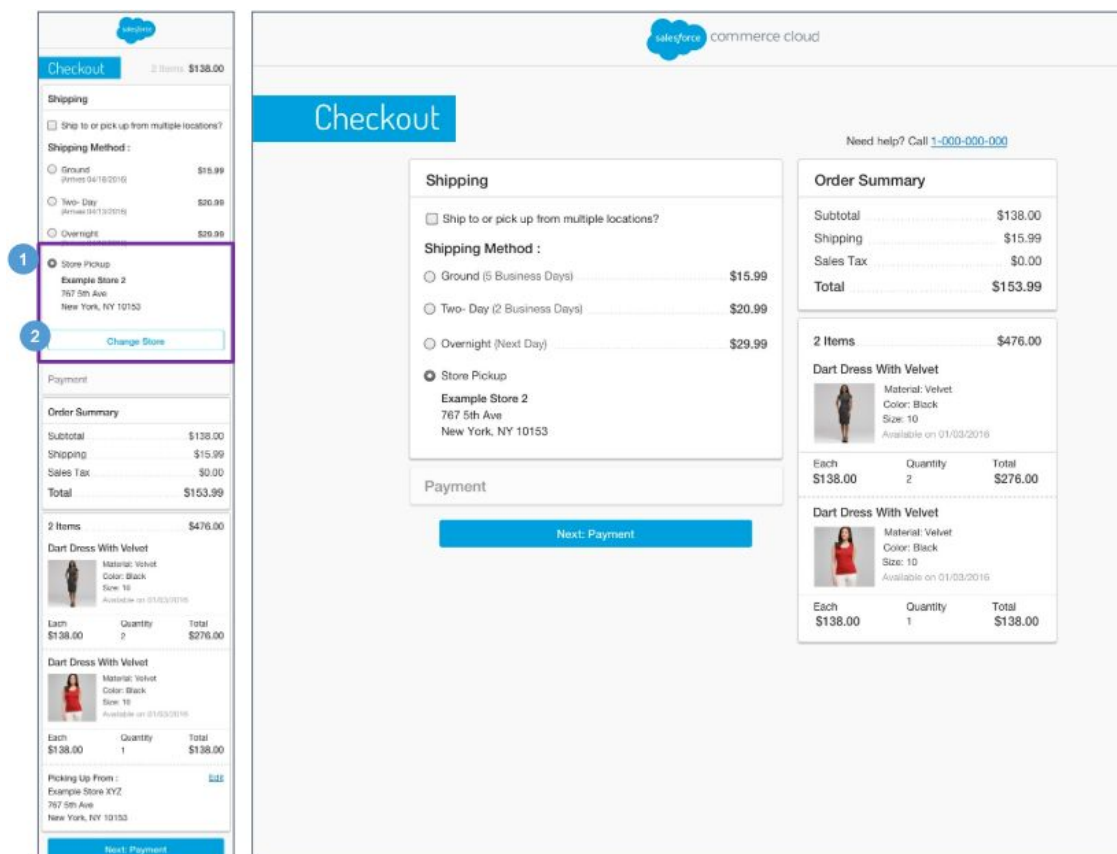
16



Mobile | Desktop:

1. The shopper can enter a zip code or a postal code.
 - a. Shows an error if an incorrect postal code or zip code is entered, or if there are no matching stores.
 - b. Shows all valid store locations in list view based on the zip code / postal code and radius. The following store attributes are shown:
 - i. Store Name
 - ii. Store Address - Clicking opens Google Maps in a browser window with the store location pinned on the map. Store attributes for latitude and longitude must be configured in Business Manager.
 - iii. Store Hours
 - iv. Phone Number - Clicking opens a prompt to dial the store's number.
2. Clicking Find Stores searches the geographical area specified.
3. Radius dropdown values are stored in a .properties file (15-300 miles), the default is 15 miles. Selecting a radius value refines the search results. Miles can be converted to kilometers via the script method 'dw.catalog.StoreMgr.searchStoresBy.'
4. The Select Store button is inactive until at least 1 store is returned in the result set.
 - a. Clicking Select Store sets the store as the preferred store for the remainder of the session.
5. This button is active even if all required fields are not filled in. Clicking Next: Payment opens the Billing & Payment Method page.
 - a. An error message is shown if the shopper clicks Next: Payment without first selecting a store by clicking Select Store.

Shipping Address - Shipping Method (Saved Store) - OOTB Sample Design:



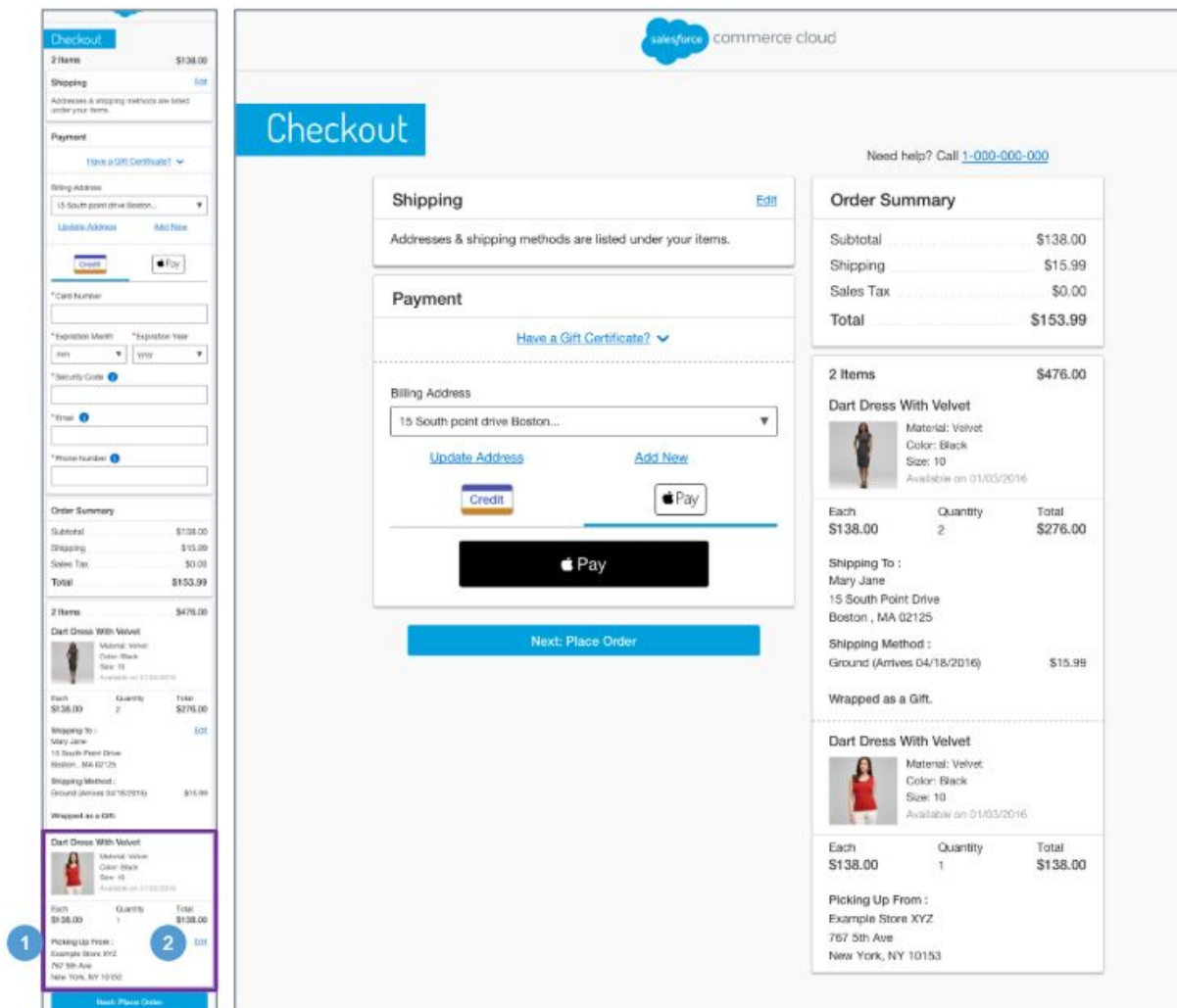
Shipping Address - Shipping Method (Saved Store) - OOTB Functional Details:

The shopper is presented this page if they have a saved store in their session.

Mobile | Desktop:

1. The selected store is shown under Shipping Method: Store Pickup. The following store attributes are shown:
 - a. Store Name
 - b. Store Address - Clicking opens Google Maps in a browser window with the store location pinned on the map. Store attributes for latitude and longitude must be configured in Business Manager.
 - c. Store Hours
 - d. Phone Number - Clicking opens a prompt to dial the store's number.
2. Clicking Change Store removes the saved store details. The Store Locator panel opens and shows the previous search.

Billing Address - Payment Method - OOTB Sample Design:



The screenshot displays the Salesforce Commerce Cloud checkout interface. The page is divided into several sections: a top navigation bar with the Salesforce logo and 'commerce cloud' text; a main heading 'Checkout'; a 'Shipping' section with an 'Edit' link; a 'Payment' section with a 'Have a Gift Certificate?' link; a 'Billing Address' section with an 'Update Address' link and an 'Add New' link; a 'Credit' button and an 'Apple Pay' button; a 'Next: Place Order' button; an 'Order Summary' section with a table of items and their prices; and a 'Shipping To' section with a table of shipping methods and their prices. The 'Billing Address' section is highlighted with a red box and a red circle, indicating it is the focus of the design.

Shipping [Edit](#)

Addresses & shipping methods are listed under your items.

Payment

[Have a Gift Certificate?](#)

Billing Address

15 South point drive Boston...

[Update Address](#) [Add New](#)

[Credit](#) [Apple Pay](#)

[Next: Place Order](#)

Order Summary

Subtotal \$138.00
Shipping \$15.99
Sales Tax \$0.00
Total \$153.99

2 Items \$476.00

Dart Dress With Velvet

Material: Velvet
Color: Black
Size: 10
Available on 01/03/2016

Each	Quantity	Total
\$138.00	2	\$276.00

Shipping To :
Mary Jane
15 South Point Drive
Boston , MA 02125

Shipping Method :
Ground (Arrives 04/18/2016) \$15.99

Wrapped as a Gift.

Dart Dress With Velvet

Material: Velvet
Color: Black
Size: 10
Available on 01/03/2016

Each	Quantity	Total
\$138.00	1	\$138.00

Picking Up From :
Example Store XYZ
767 5th Ave
New York, NY 10153

Billing Address - Payment Method - OOTB Functional Details:

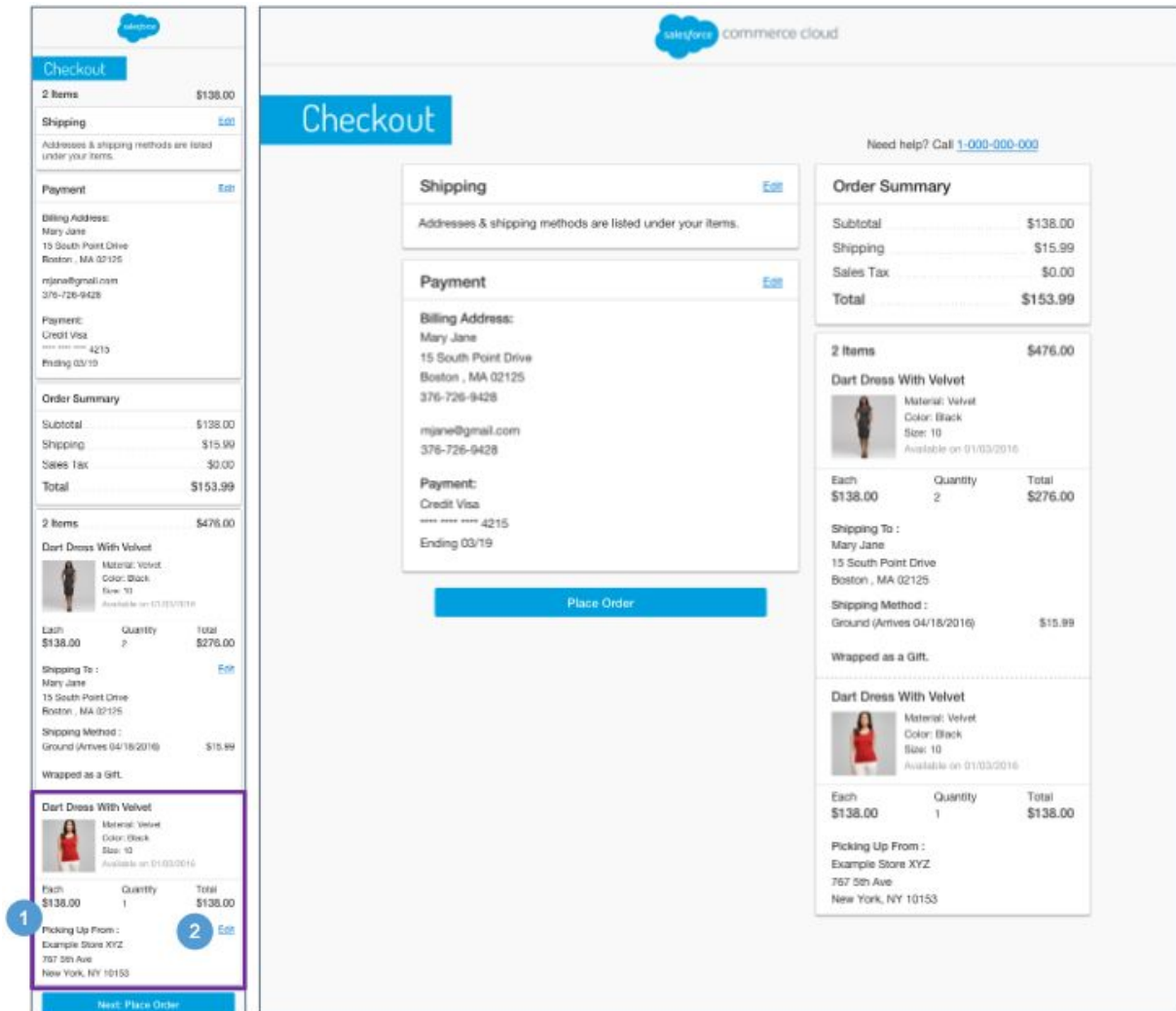
The Billing Address | Payment Method page for a product with in-store pickup has the same functionality as the Billing Address | Payment Method for a standard product or a variation product with the following exceptions.

Mobile | Desktop:

1. Shows store attributes:
 - a. Store Name
 - b. Store Address
 - c. Phone Number

- Clicking Edit opens the Store Locator page, enabling the shopper to perform a new store search.

Place Order - OOTB Sample Design:



Checkout

2 Items \$138.00

Shipping [Edit](#)

Addresses & shipping methods are listed under your items.

Payment [Edit](#)

Billing Address:
Mary Jane
15 South Point Drive
Boston, MA 02125
mjane@gmail.com
376-726-9428

Payment:
Credit Visa
www.***.*** 4215
Ending 03/19

Order Summary

Subtotal	\$138.00
Shipping	\$15.99
Sales Tax	\$0.00
Total	\$153.99

2 Items \$476.00

Dart Dress With Velvet

Material: Velvet
Color: Black
Size: 10
Available on 01/03/2016

Each	Quantity	Total
\$138.00	2	\$276.00

Shipping To:
Mary Jane
15 South Point Drive
Boston, MA 02125

Shipping Method:
Ground (Arrives 04/18/2016) \$15.99

Wrapped as a Gift.

Dart Dress With Velvet

Material: Velvet
Color: Black
Size: 10
Available on 01/03/2016

Each	Quantity	Total
\$138.00	1	\$138.00

Picking Up From:
Example Store XYZ
767 5th Ave
New York, NY 10153

[Next: Place Order](#)

Place Order - OOTB Functional Details:

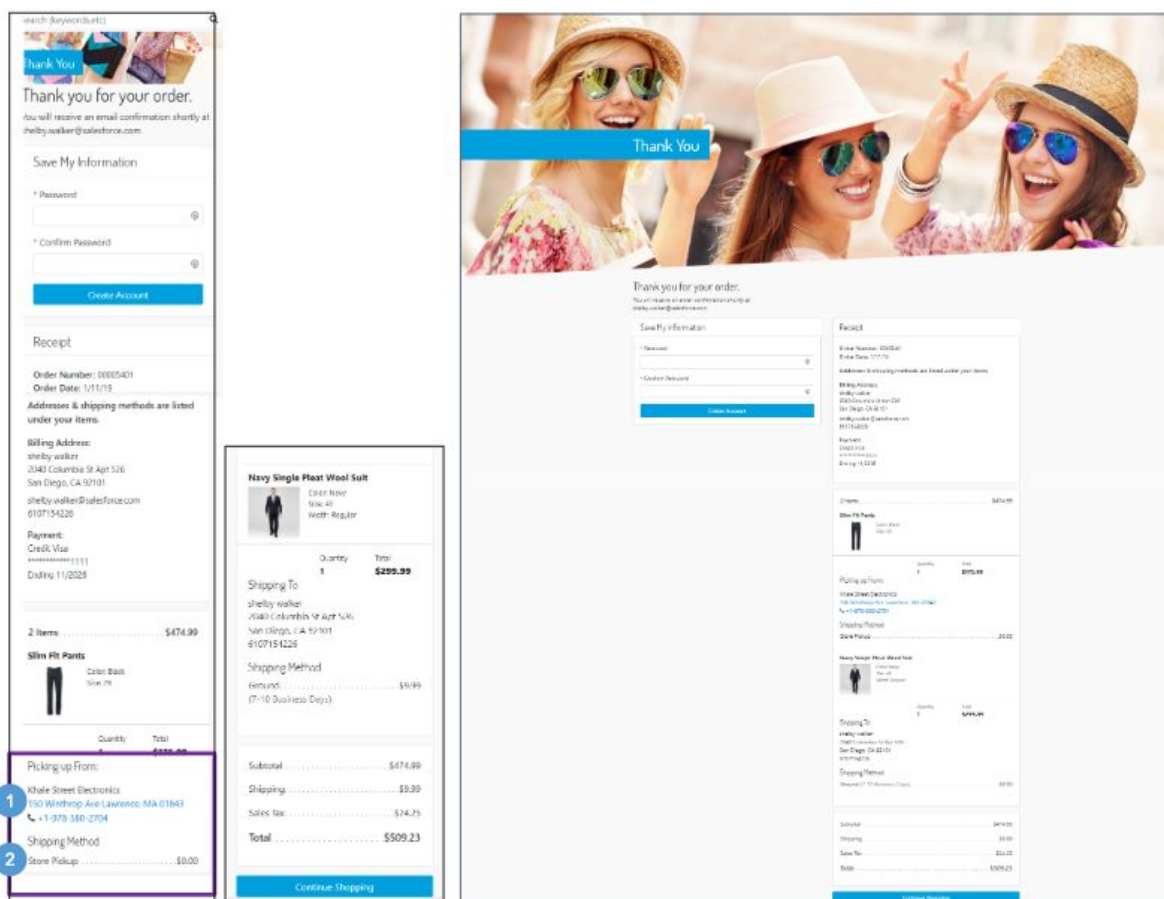
The Place Order page for a product with in-store pickup has the same functionality as the Place Order page for a standard product or a variation product with the following exceptions.

Mobile | Desktop:

- Shows store attributes:
 - Store Name
 - Store Address
 - Phone Number

2. Clicking Edit opens the Store Locator page, enabling the shopper to perform a new store search.

Order Receipt - OOTB Sample Design:



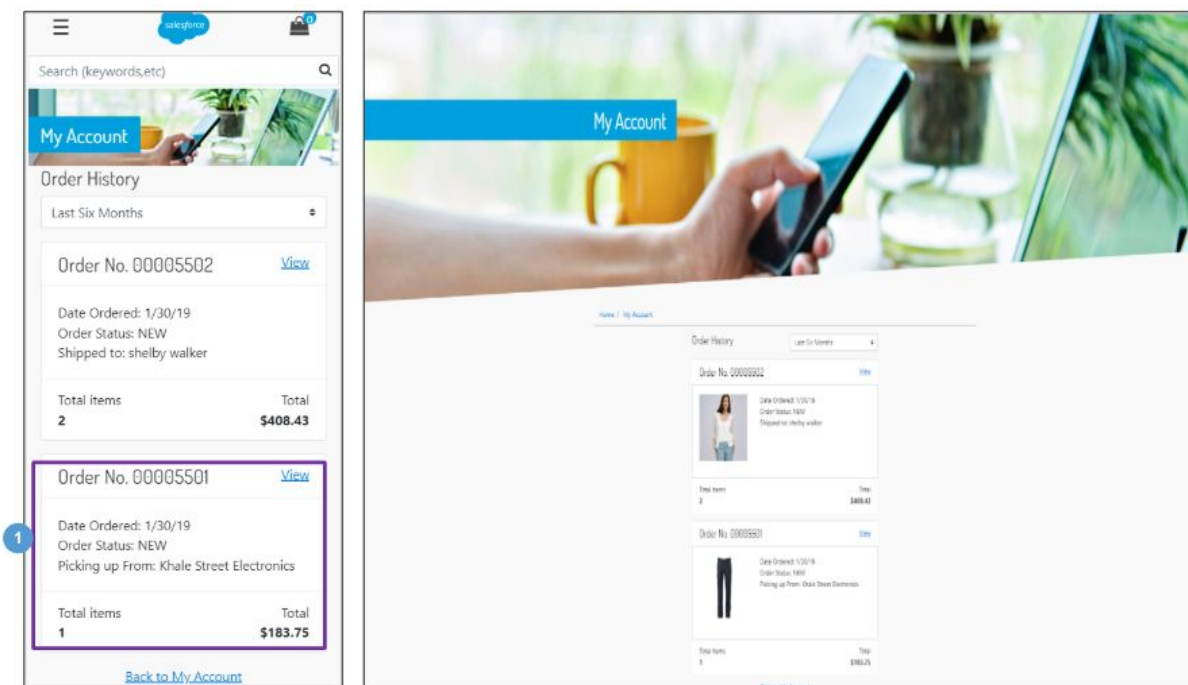
Order Receipt - OOTB Functional Details:

The Order Receipt page for a product with in-store pickup has the same functionality as the Order Receipt page for a standard product or a variation product with the following exceptions.

Mobile | Desktop:

1. Shows store attributes:
 - a. Store Name
 - b. Store Address - Clicking opens Google Maps in a browser window with the store location pinned on the map. Store attributes for latitude and longitude must be configured in Business Manager.
 - c. Phone Number - Clicking opens a prompt to dial the store's number.
2. Shipping Method is shown as Store Pickup with a \$0.00 shipping charge.

Order History - OOTB Sample Design:



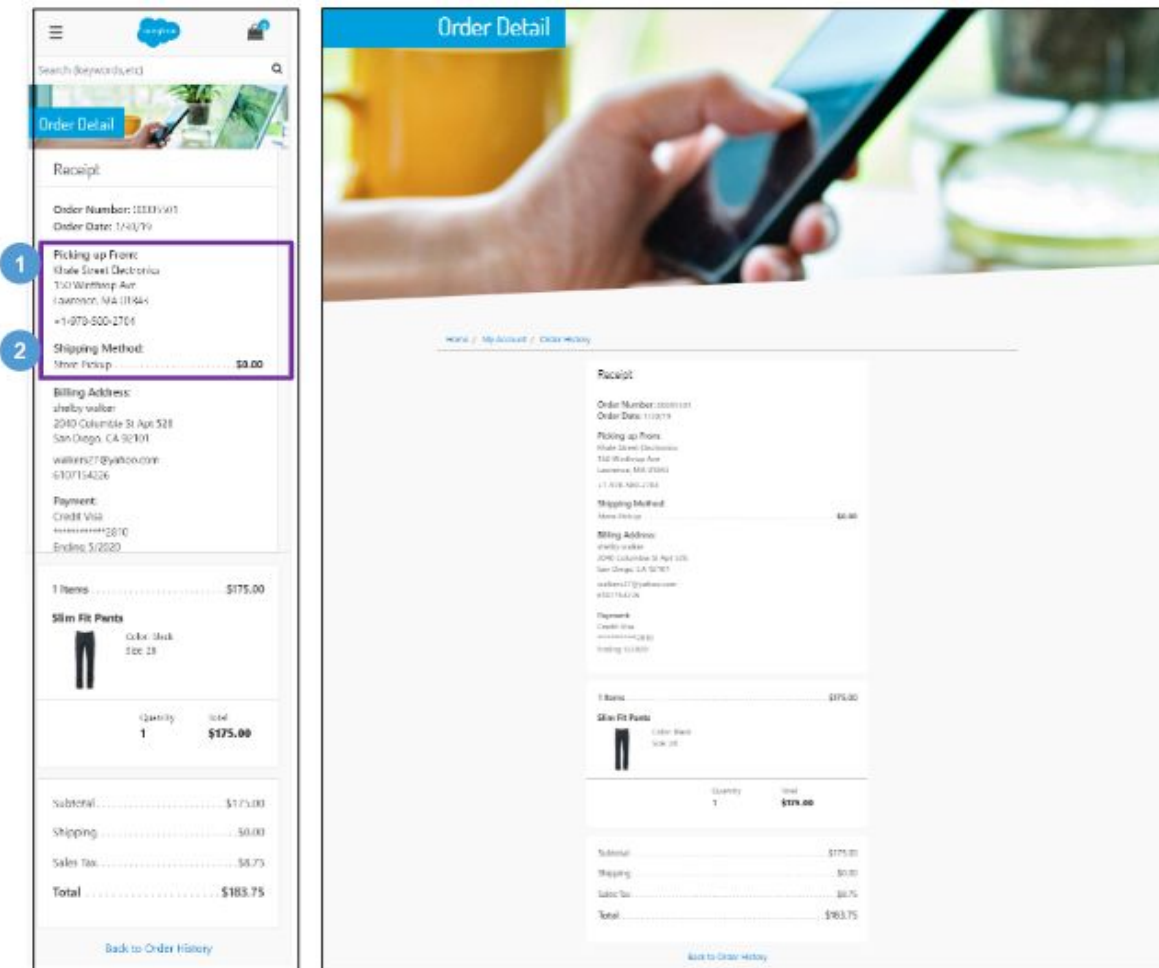
Order History - OOTB Functional Details:

The Order History page for a product with in-store pickup has the same functionality as the Order History page for a standard product or a variation product with the following exceptions.

Mobile | Desktop:

1. Shows a compact view of the store details 'Picking up from: [store name].'

Order Detail - OOTB Sample Design:



Order Detail - OOTB Functional Details:

The Order Detail page for a product with in-store pickup has the same functionality as the Order Detail page for a standard product or a variation product with the following exceptions.

Mobile | Desktop:

- Shows store attributes:
 - Store Name
 - Store Address.
 - Phone Number
- Shipping Method is shown as Store Pickup with a \$0.00 shipping charge.

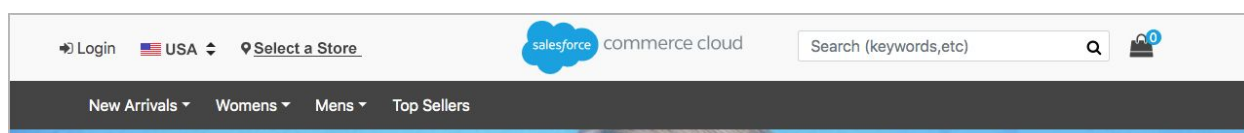
5. Test your solution!

As with any enhancements you make to your storefront, follow your organization's process for testing thoroughly to ensure your data is accurate, data flows across all systems are operating as expected, impacted teams know of any changes to their process, and all storefront functionality works as expected, before launching your new BOPIS offering.

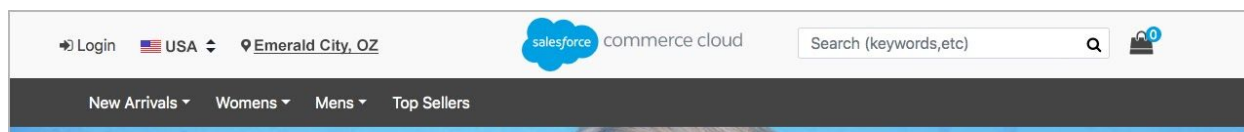
Optional Additional Enhancements

There are a few other enhancements you may wish to make to your storefront on top of what is included with OOTB functionality, to enhance the BOPIS shopper experience. And while these are not required for BOPIS functionality, we've provided some information here on the most common enhancements we've seen merchants use.

1. Surface the **Store Locator** in your Header
 - a. This placement draws attention to the Store Locator, and allows the shopper to search for and select a store before they browse to a PDP.
 - b. Clicking on the Store Locator link could open a modal window, or direct to the Store Locator page.
 - i. **Technical Tip:** The selected store ID can be stored in a cookie and the selected store details can be loaded into the header via an AJAX request and data injection.
 - c. Below is a simple example of what this could look like, before and after a store is selected.

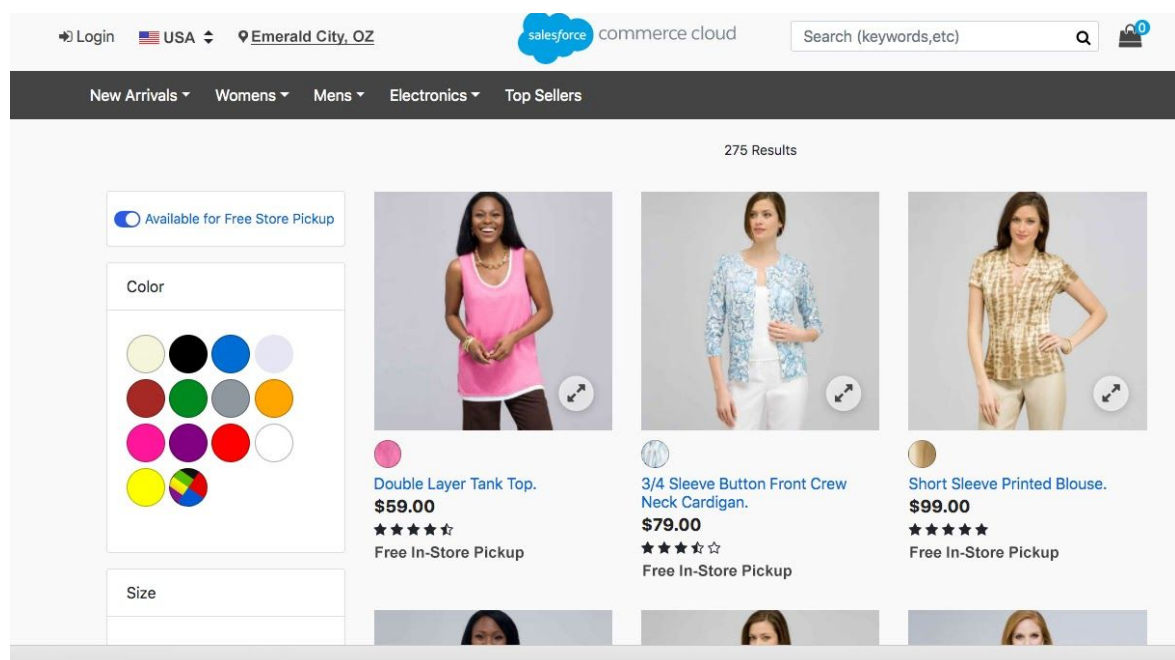


With store selected:



2. Add a filter on your Product Listing Page for Products identified as Available for Store Pickup
 - a. Add this filter to be shown along with other PLP refinements that are managed in Business Manager.

- b. On the product tiles, use the custom attribute to trigger the addition of a callout message to signify a product is one that could be available for BOPIS, such as “Free In-Store Pickup”
 - i. **Technical Tip:** This customization uses the **availableForInStorePickup** attribute on products to determine whether a flag or callout message appears on a product tile to denote that an item is sold in stores. This is not a check for store inventory availability however, as that check will happen only on the PDP, once a SKU has been selected.




3. Add a form field entry in Checkout for the shopper to specify an additional person to pick up the order.
 - a. **Technical Tip:** This would require an update to your Order Export file, to add the additional data attributes.

1 Contact Information [Edit](#)


2 Store Pickup

Items to Pick Up



Ruby Slippers
Item#
Price: \$19.39
Quantity: 1

Pick Up Location


1 Yellow Brick Road
Elmarded City, OZ

☒ Add an additional pickup person (optional)

First Name (required)

Last Name (required)

Email Address (required)

We'll send an email to you and your additional pick up person once your order is ready.

NEXT STEP >

Order Summary

Order Subtotal:	\$19.39
Estimated Shipping	FREE
Estimated Tax [?]	TBD
Estimated Total:	\$19.39

Store Pick-up Considerations

The process for how you route orders to your stores will depend on your specific systems and implementation.

- You may have an in-store app that Store Associates can use to receive orders and mark them as ready for pick-up.
- You may have a way to access your OMS or ERP system in the store, to view and update orders
- Or you may have a more basic process for getting the order information to your stores for Associates to pick and pack.

Regardless of your method, you'll want to develop a clear process for your Store Associates for handling order pick up, and how to deal with customer order inquiries, returns, etc. You'll also need a way to let customers know that the orders are Ready for Pick up, and, once picked up, to close the loop on the order process and mark the orders as fulfilled.

Offering Curbside Pick up? Here are some additional considerations to maintain customer satisfaction and safety for everyone:

- Have clear signage for where customers should wait
- Provide a store phone number they can 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 pick up.
- Have dedicated staff to communicate with customers and to take the packages to their car.



Additional Resources

Ask the Community

Get answers to your questions from B2C Commerce experts and other customers via the [B2C Commerce Trailblazer Community Group](#)

Review additional Documentation

Search and find specifications and technical info for all aspects of B2C Commerce on the [Salesforce B2C Commerce Infocenter](#).

1-on-1 Help

Request a [Commerce Cloud Accelerator](#)

Upcoming Events & Webinars

Review the [Calendar](#) of upcoming Commerce Cloud learning opportunities

Engage a Certified Commerce Cloud Architect

Need some expert guidance for planning your BOPIS solution? Reach out to your Customer Success Manager to ask about Advisory Services.