



Extend Integration for Salesforce Commerce Cloud SiteGenesis

Welcome to Extend! We provide an easy way for any merchant to sell extended warranties - generating new revenue, increasing purchase conversion, and dramatically improving the customer experience.

This guide will walk you through the process of getting live with Extend Protection Plans, so you can start selling and earning revenue as quickly and easily as possible. The process is pretty simple - and even more simple with a connected SFCC SiteGenesis v19.10.0 store. Read on to learn more!



Table of Contents

Extend Integration for Salesforce Commerce Cloud SiteGenesis	1
Table of Contents	2
Component Overview	3
Functional Overview	3
Use Cases	3
Compatibility	3
Privacy Policies on Payments	3
Implementation Guide	5
Setup of Business Manager	5
Upload cartridge & include in the cartridge path	5
Import metadata	5
Import custom jobs	5
Import service	5
Import Extend catalog	6
Configuration	6
Custom preferences	6
Jobs	7
Extend Products Export	7
Extend Contracts Creation	8
Custom Code	9
External Interfaces	30
Firewall Requirements	30
Orders API	31
Custom preferences	31
Extend SDK analytics	36
Custom Preferences	36
Testing	41
Enable Extend	41
PDP	41
Cart	43
Business Manager	45
Questions? Feedback?	46





Component Overview

1. Functional Overview

This integration cartridge enables the following features:

- Daily product catalog export to Extend
- Present offers via the Product Display Page (PDP), an up-sell modal during the add-to-cart action, and within the Cart page for eligible products
- Purchase Extend Protection Plans for eligible products
- Contract creation and cancellation with Extend via the API

2. Use Cases

For a merchant, the primary use cases include:

- Maintaining core Extend functionalities directly within Business Manager
- Display Extend Protection Plan products in PDP & Cart
- Display an Extend Protection Plan up-sell modal in PDP & Cart
- Capture customer behavior with Extend's analytics module
- Generate contracts for newly purchased protection plans with Extend
- Cancel contracts for returned orders with Extend

For a customer, the primary use cases include:

- Attach an Extend Protection Plan to an eligible catalog product during their purchase journey
- View details, including coverage and terms & conditions about the Extend Protection Plans
- Remove Extend Protection Plans from Cart (either by removing the main product or just the warranty product) during their purchase journey

3. Compatibility

This cartridge has been developed and tested against Commerce Cloud Digital 19.10 and integrated on top of SiteGenesis 104.1.5.

4. Privacy Policies on Payments

This cartridge implements a custom object for temporary storage of order data needed for Extend Protection Plan contract generation. This custom object (ExtendContractsQueue) was implemented based on a queue pattern that is consumed



by a job configured to execute every 5 minutes. This data is not exposed publicly and used only in a server-to-server communication setup.



Extend does not handle payments from the end-user. The merchant is responsible for charging the end-user for the Extend Protection Plans purchased, using the existing tenders that are currently supported on the specific environment. The merchant is responsible for generating Extend Protection Plan contracts only after charging the end-user for the warranty plans purchased. Please visit to <https://extend.com/privacy> for more information.





Implementation Guide

The SiteGenesis version of this integration follows the best practices for this storefront version. In order to achieve the features described in this document, please use the sample code snippets below as reference for this integration.

1. Setup of Business Manager

1.1. Upload cartridge & include in the cartridge path

- i. Upload the int_extend cartridge to the sandbox.
- ii. Go to Business Manager → Administration → Site → Manage Sites. Select the desired site and go to Settings. At the beginning of the Cartridge Path add “int_extend:” in order to include the cartridge in the current site’s configuration.

1.2. Import metadata

- i. Go to Business Manager → Administration → Site Development → Import & Export
- ii. Upload the following files from ./metadata/meta/ folder
 - system-objecttype-extensions.xml
 - custom-objecttype-definitions.xml
- iii. Go to Business Manager → Administration → Site Development → Import & Export → Meta Data → Import. Both previous files should be available for import. Proceed to importing both.

1.3. Import custom jobs

- i. Go to Business Manager → Administration → Operations → Import & Export
- ii. Upload the following file: ./metadata/jobs.xml
- iii. Go to Business Manager → Administration → Operations → Import & Export → Jobs → Import. The previous file should be available for import. Proceed to importing it.

1.4. Import service

- i. Go to Business Manager → Administration → Operations → Import & Export
- ii. Upload the following file: ./metadata/services.xml
- iii. Go to Business Manager → Administration → Operations → Import & Export → Services → Import. The previous file should be available for import. Proceed to importing it.





- iv. Note: by default, the service URL is configured to point to the demo instance of the Extend API. Prior to going live with the integration, this will need to be changed and tested with the live production API.

1.5. Import Extend catalog

- i. Select the site integrating with Extend
- ii. Go to Business Manager → Merchant Tools → Products and Catalogs → Import & Export
- iii. Upload the following file: ./metadata/catalog.xml. This will import 3 dummy option products, which are used in order to create dynamic product line items in the cart. We recommend keeping this as a separate master catalog, for visibility purposes.

2. Configuration

2.1. Catalog setup

- i. The 3 Extend products that require configuration are:
 - EXTEND-12
 - EXTEND-24
 - EXTEND-36
- ii. To make the Extend products merchandisable, perform the following:
 - Assign to the storefront catalog
 - Allocate an inventory record in the inventory file assigned to the current site
 - Configure this entry as perpetual, as these are intangible goods and need to always be in-stock
- iii. These Extend products do not require any description, pricing, or any other regular product attributes to be configured

2.2. Custom preferences

- i. Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → Extend
- ii. Configure the provided Extend credentials in the following fields
 - **Access Token:** this is used for authenticating web service calls to Extend API. Note: this is available from the Extend Merchant Portal and is unique to the environment (Demo and Production)
 - **Store ID:** this is used for the client-side Javascript SDK to identify the current store. Note: this is available from the Extend Merchant Portal and is unique to the environment (Demo and Production)





- iii. Configure the image breakpoint for using when exporting the product feed
 - **Image View Type** is by default configured to large
 - Configure this to specify the breakpoint for the largest image size available across your products
- iv. **Extend SDK**
By default, this points to the Extend Javascript SDK, [more info here](#)
- v. **Use Extend in Production Mode**
By default, this is set to “No”, which points the integration to the Extend Demo environment. During production cutover, please ensure this is set to “Yes”
- vi. **Extend API Method**
By default, the cartridge leverages Extend’s new Orders API, which enables additional features such as leadTokens generation and analytics. To use the previous API version, choose Contracts API (please confirm with the Extend project team before changing versions)
- vii. **Extend API Version**
By default, this points to the ‘Default’, Please set this to the latest available version in the option.
- viii. Use the provided feature switches in order to enable / disable Extend features
 - **Enable Extend:** Enables / disables Extend globally
 - **Enable in PDP:** Enables / disables Extend in PDP
 - **Enable Up-sell in PDP:** Enables / disables the up-sell modal in PDP
 - **Enable Up-sell in Cart:** Enables / disables the up-sell button and modal in Cart
 - **Extend Analytics Switch:** Enables / disables Extend’s analytics module. The analytics module enables revenue metrics such as attach rate
 - **Extend Secret Key:** used as part of Extend’s contract cancellation process in SFCC if applicable

3. Jobs

3.1. Extend Products Export

This job uses Extend’s API to traverse the storefront catalog and export each Simple and Variant Product that is, at the time of the job run, merchandisable (online and in stock).

- i. Go to Business Manager → Operations → Jobs. Select the Extend Products Export job.
- ii. Go to Job Steps and select Scope. Configure this to run in the scope of the site integrating with Extend.





- iii. Schedule this to run once a day, at a time that's most suitable for a recurring job setup, ideally after the catalog is being updated.
- iv. Ensure this job is triggered at least once without errors and notify Extend after the successful run. Extend will confirm if the product sync was successful.

3.2. Extend Contracts Creation

This job uses Extend's Contracts API to consume the contracts queue, stored in the ExtendContractsQueue custom object. During SFCC order creation (after checkout), an instance of this object is created for each Extend Protection Plan on the SFCC order. When this job is triggered, it sends a contract generation request to Extend for each queued element. Upon successful contract creation, it removes the completed instance from the queue. Note: this job is not required when using [the Orders API](#) as contracts are created immediately during SFCC order creation for eligible line items.

3.3. Create Refunds from SFCC

This job uses Extend's API to cancel an active contract on a cancelled SFCC order (Order Status = "Cancelled"). Please note that Extend will never refund the customer directly, as Extend does not capture the customer's payment information at any point. Customer refunds should be processed based on the merchant's regular refund process. This job will simply terminate active contract(s) after the customer's refund request is accepted by the merchant. All cancelled Extend contracts are represented as credit lines on Extend's monthly invoice. For additional information on Extend's refund policy, please reach out to the Extend project team. This job is not required if the merchant's return and refund process occurs outside of SFCC as the contract cancellation requests to Extend will likely occur in a downstream system. Confirm with the Extend project team prior to scheduling this job.





4. Custom Code

4.1. Include extend_footer.isml

- In:
 * /app_storefront_core/cartridge/templates/default/components/footer/footer_UI.isml
- Where: at the end of the file

Code:

```
<iscomment>Extend Integration</iscomment>
<isinclude template="components/footer/extend_footer"/>
```

```
1  <iscontent type="text/html" charset="UTF-8" compact="true"/>
2
3  <!--[if gte IE 9 | !IE]><!-->
4  |  <script src="${URLUtils.staticURL('/lib/jquery/jquery-2.1.1.min.js')}" type="text/javascript"></script>
5  <!--<![endif]-->
6
7  <!--[if lte IE 8]>
8  |  <script src="${URLUtils.staticURL('/lib/jquery/jquery-1.11.1.min.js')}" type="text/javascript"></script>
9  |  <script src="//cdn.rawgit.com/weblinc/media-match/master/media.match.min.js" type="text/javascript"></script>
10 |  <script src="//cdnjs.cloudflare.com/ajax/libs/es5-shim/3.4.0/es5-shim.min.js"></script>
11 <![endif]-->
12
13 <!--[if IE 9]>
14 |  <script src="//cdn.rawgit.com/paulirish/matchMedia.js/master/matchMedia.js" type="text/javascript"></script>
15 |  <script src="//cdn.rawgit.com/paulirish/matchMedia.js/master/matchMedia.addListener.js" type="text/javascript"></script>
16 <![endif]-->
17
18 <script src="${URLUtils.staticURL('/lib/jquery/ui/jquery-ui.min.js')}" type="text/javascript"></script>
19
20 <iscomment>third-party add-ons</iscomment>
21 <script src="${URLUtils.staticURL('/lib/jquery/jquery.jcarousel.min.js')}" type="text/javascript"></script>
22 <script src="${URLUtils.staticURL('/lib/jquery/jquery.validate.min.js')}" type="text/javascript"></script>
23 <script src="${URLUtils.staticURL('/lib/jquery/jquery.zoom.min.js')}" type="text/javascript"></script>
24 <script type="text/javascript"><isinclude template="resources/appresources"/></script>
25 <script type="text/javascript"><isinclude url="${URLUtils.url('Resources-LoadTransient')}"/></script>
26 <script>var consent = ${session.custom.consentTracking};</script>
27 <script src="${URLUtils.staticURL('/js/app.js')}"></script>
28 <isif condition="${!('pageContext' in this) || empty(pageContext)}">
29 |  <isscript>pageContext = new Object();</isscript>
30 </isif>
31 <script>pageContext = <isprint value="${JSON.stringify(pageContext)}" encoding="off"/>;</script>
32 <script>
33 var meta = "${pdict.CurrentPageMetaData.description}";
34 var keywords = "${pdict.CurrentPageMetaData.keywords}";
35 </script>
36
37 <iscomment>Extend Integration</iscomment>
38 <isinclude template="components/footer/extend_footer"/>
```





4.2. Include extend_productcontent.isml

- In: */app_storefront_core/cartridge/templates/default/product/productcontent.isml
- Where: after the availability section, before the quantity section

Code:

```
<iscomment>
  Extend Integration
=====
</iscomment>
<isinclude template="product/extend_productcontent" />
```

```
137   <div class="availability-instore">
138     <i class="fa fa-briefcase fa-lg pull-left"></i>
139     <label for="Stock">${Resource.msg('product.instorestock','product',null)}</label>
140     <isif condition="${empty(pdct.CurrentHttpParameterMap.uuid.value)}">
141       <div id="${pdct.Product.ID}" class="availability-results availability-msg store-stock">
142         <span class="label set-preferred-store"><a href="${URLUtils.url('StoreInventory-SetZipCodeCore')}</a></span>
143       </div>
144     </isif>
145   </div>
146   <iselse/>
147   <div class="availability-web">
148     <label for="Stock">${Resource.msg('global.availability','locale',null)}</label>
149     <isif condition="${!pdct.Product.master && !pdct.Product.variationGroup}">
150       <span class="value"><isinclude template="product/components/availability"/></span>
151     </isif>
152   </div>
153   <iselse/>
154   <div class="availability-novariation">${Resource.msg('product.selectforstock','product',null)}</div>
155 </isif>
156 </div>
157 </iselse>
158 </div>
159 <iscomment>
160   Extend Integration
161   =====
162 </iscomment>
163 <isinclude template="product/extend_productcontent" />
164 <iscomment>
165   product quantity
166   =====
167 </iscomment>
```





4.3. Include PDP functionality

- In: */app_storefront_core/cartridge/js/pages/product/variant.js
- Where: function updateContent(), at the end of the \$.ajax() callback function

Code:

```
if (window.EXT_GLOBAL_SWITCH) {  
    window.extendPDP();  
}
```

```
1  'use strict';  
2  
3  var ajax = require('....ajax'),  
4      image = require('./image'),  
5      progress = require('....progress'),  
6      productStoreInventory = require('....storeinventory/product'),  
7      tooltip = require('....tooltip'),  
8      util = require('....util');  
9  
10  
11  /**  
12   * @description update product content with new variant from href, load new content to #product-content panel  
13   * @param {String} href - url of the new product variant  
14  **/  
15  var updateContent = function (href) {  
16      var $pdpForm = $('.pdpForm');  
17      var qty = $pdpForm.find('input[name="Quantity"]').first().val();  
18      var params = {  
19          Quantity: isNaN(qty) ? '1' : qty,  
20          format: 'ajax',  
21          productlistid: $pdpForm.find('input[name="productlistid"]').first().val()  
22      };  
23  
24      progress.show($('#pdpMain'));  
25  
26      ajax.load({  
27          url: util.appendParamsToUrl(href, params),  
28          target: $('#product-content'),  
29          callback: function () {  
30              if (SitePreferences.STORE_PICKUP) {  
31                  productStoreInventory.init();  
32              }  
33              image.replaceImages();  
34              tooltip.init();  
35              if (window.EXT_GLOBAL_SWITCH) {  
36                  window.extendPDP();  
37              }  
38          }  
39      });  
40  };
```





4.4. Include Upsell / cross-sell modal functionality

- In: */app_storefront_core/cartridge/js/pages/product/addToCart.js
- Where: function addToCart(), after the \$form variable declaration and before addItemToCart()

Code:

```
// Extend Integration
if ($('#extend-offer').length) {
    extendAddToCart($form, page, minicart, dialog, addCartItem);
    return;
}
```

```
35  /**
36   * @description Handler to handle the add to cart event
37   */
38 var addToCart = function (e) {
39     e.preventDefault();
40     var $form = $(this).closest('form');
41
42     // Extend Integration
43     if ($('#extend-offer').length) {
44         extendAddToCart($form, page, minicart, dialog, addCartItem);
45         return;
46     }
47
48     addCartItem($form).then(function (response) {
49         var $uuid = $form.find('input[name="uuid"]');
50         if ($uuid.length > 0 && $uuid.val().length > 0) {
51             page.refresh();
52         } else {
53             // do not close quickview if adding individual item that is part of product set
54             // @TODO should notify the user some other way that the add action has completed successfully
55             if (!$(this).hasClass('sub-product-item')) {
56                 dialog.close();
57             }
58             minicart.show(response);
59         }
60     }).bind(this));
61 };
62 
```





4.5. Import extendHelpers.js in CartModel.js

- In: */app_storefront_controllers/cartridge/scripts/models/CartModel.js
- Where: include this script at the API Includes section

Code:

```
// Import Extend Script Helpers
var extendHelpers = require('*cartridge/scripts/extend');
```

```
1  'use strict';
2  /**
3   * Model for cart functionality. Creates a CartModel class with payment, shipping, and product
4   * helper methods.
5   * @module models/CartModel
6   */
7  var Transaction = ...require('dw/system/Transaction');
8
9  /* API Includes */
10 var AbstractModel = require('./AbstractModel');
11 var ArrayList = require('dw/util/ArrayList');
12 var BasketMgr = require('dw/order/BasketMgr');
13 var Money = require('dw/value/Money');
14 var MultiShippingLogger = dw.system.Logger.getLogger('multishipping');
15 var OrderMgr = require('dw/order/OrderMgr');
16 var PaymentInstrument = require('dw/order/PaymentInstrument');
17 var Product = require('~cartridge/scripts/models/ProductModel');
18 var ProductInventoryMgr = require('dw/catalog/ProductInventoryMgr');
19 var ProductListMgr = require('dw/customer/ProductListMgr');
20 var QuantityLineItem = require('~cartridge/scripts/models/QuantityLineItemModel');
21 var Resource = require('dw/web/Resource');
22 var ShippingMgr = require('dw/order/ShippingMgr');
23 var StoreMgr = require('dw/catalog/StoreMgr');
24 var TransientAddress = require('~cartridge/scripts/models/TransientAddressModel');
25 var UUIDUtils = require('dw/util/UUIDUtils');
26
27 var lineItem;
28 var app = require('~cartridge/scripts/app');
29 var ProductList = app.getModel('ProductList');
30
31 // Import Extend Helpers
32 var extendHelpers = require('*cartridge/scripts/extendHelpers');
```





4.6. Include Extend line item creation in CartModel.js

- In: */app_storefront_controllers/cartridge/scripts/models/CartModel.js
- Where: function addProductToCart(), before the return statement

Code:

```
// Extend Create Warranty LineItem
extendHelpers.createExtendLineItem(cart, params, Product, Transaction);
```

```
106      // Adds a product.
107  } else {
108      var previousBonusDiscountLineItems = cart.getBonusDiscountLineItems();
109      productToAdd = Product.get(params.pid.stringValue);
110
111      if (productToAdd.object.isProductSet()) {
112          var childPids = params.childPids.stringValue.split(',');
113          var childQtys = params.childQtys.stringValue.split(',');
114          var counter = 0;
115
116          for (var i = 0; i < childPids.length; i++) {
117              var childProduct = Product.get(childPids[i]);
118
119              if (childProduct.object && !childProduct.isProductSet()) {
120                  var childProductOptionModel = childProduct.updateOptionSelection(params);
121                  cart.addProductItem(childProduct.object, parseInt(childQtys[counter]), childProductOptionModel);
122              }
123              counter++;
124          }
125      } else {
126          productOptionModel = productToAdd.updateOptionSelection(params);
127          cart.addProductItem(productToAdd.object, params.Quantity.doubleValue, productOptionModel);
128      }
129
130      // When adding a new product to the cart, check to see if it has triggered a new bonus discount line item.
131      newBonusDiscountLineItem = cart.getNewBonusDiscountLineItem(previousBonusDiscountLineItems);
132  }
133
134  // Extend Create Warranty LineItem
135  extendHelpers.createOrUpdateExtendLineItem(cart, params, Product);
136
137  return {
138      format: format,
139      template: template,
140      BonusDiscountLineItem: newBonusDiscountLineItem
141  };
142},
143
```





4.7. Import extendHelpers.js in Cart

- In: */app_storefront_controllers/cartridge/controllers/Cart.js
- Where: include this script at the API Includes section

Code:

```
var extendHelpers = require('*cartridge/scripts/extendHelpers');
```

```
1  'use strict';
2
3  /**
4   * Controller that adds and removes products and coupons in the cart.
5   * Also provides functions for the continue shopping button and minicart.
6   *
7   * @module controllers/Cart
8   */
9
10 /* API Includes */
11 var ArrayList = require('dw/util/ArrayList');
12 var ISML = require('dw/template/ISML');
13 var Resource = require('dw/web/Resource');
14 var Transaction = require('dw/system/Transaction');
15 var URLUtils = require('dw/web/URLUtils');
16
17 /* Script Modules */
18 var app = require('~/cartridge/scripts/app');
19 var guard = require('~/cartridge/scripts/guard');
20
21 /* EXTEND */
22 var extendHelpers = require('*cartridge/scripts/extendHelpers');
```





4.8. Add functionality to remove Extend warranty line items in Cart.js

- In: */app_storefront_controllers/cartridge/controllers/Cart.js
- Where: ‘deleteProduct’ action handler; replace this entire function with the following code

Code:

```
'deleteProduct': function (formgroup) {
    // EXTEND - Check if there is a warranty associated with the product
    // that is being deleted
    var warrantyLineItem = extendHelpers.checkForWarrantyLI(cart,
    formgroup.getTriggeredAction().object);
    Transaction.wrap(function () {
        cart.removeProductLineItem(formgroup.getTriggeredAction().object);

        // EXTEND - if there is a warranty associated with the product
        remove the corresponding warranty
        if (warrantyLineItem) {
            cart.removeProductLineItem(warrantyLineItem);
        }
    });
    return {
        cart: cart
    };
}
```

```
168  'deleteGiftCertificate': function (formgroup) {
169      Transaction.wrap(function () {
170          cart.removeGiftCertificateLineItem(formgroup.getTriggeredAction().object);
171      });
172
173      return {
174          cart: cart
175      };
176  },
177  'deleteProduct': function (formgroup) {
178
179      // EXTEND - Check if there is a warranty associated with the product that is being deleted
180      var warrantyLineItem = extendHelpers.checkForWarrantyLI(cart, formgroup.getTriggeredAction().object);
181
182      Transaction.wrap(function () {
183          cart.removeProductLineItem(formgroup.getTriggeredAction().object);

184          // EXTEND - if there is a warranty associated with the product remove the corresponding warranty
185          if (warrantyLineItem) {
186              cart.removeProductLineItem(warrantyLineItem);
187          }
188      });
189
190      return {
191          cart: cart
192      };
193  },
194  'editLineItem': function (formgroup) {
```





4.9. Disable edit in cart for Extend line items

- In: */app_storefront_core/cartridge/templates/default/checkout/cart/cart.isml
- Where: line 176: <iscomment>Call module to render product</iscomment>

Replace:

```
<isdisplayliproduct p_productli="${lineItem}" p_formli="${FormLi}"  
p_editable="${true}" p_hideprice="${true}" p_hidepromo="${false}" />
```

With:

```
<isif condition="${lineItem.custom.parentLineItemUUID}">  
    <isdisplayliproduct p_productli="${lineItem}" p_formli="${FormLi}"  
p_editable="${false}" p_hideprice="${true}" p_hidepromo="${false}" />  
<iselse>  
    <isdisplayliproduct p_productli="${lineItem}" p_formli="${FormLi}"  
p_editable="${true}" p_hideprice="${true}" p_hidepromo="${false}" />  
</isif>
```

```
163      <iscomment> Skip bonus-choice products in first pass. </iscomment>  
164      <isif condition="${lineItem.bonusDiscountLineItem == null}">  
165          <tr class="cart-row" data-uuid="${lineItem.getUUID()}">  
166              <td class="item-image">  
167                  <isif condition="${lineItem.product != null && lineItem.product.getImage('small',0) != null}">  
168                        
170                            
171                  </isif>  
172              </td>  
173              <td class="item-details">  
174                  <iscomment>Call module to render product</iscomment>  
175                  <isif condition="${lineItem.custom.parentLineItemUUID}">  
176                      <isdisplayliproduct p_productli="${lineItem}" p_formli="${FormLi}" p_editable="${false}" p_hideprice="${true}" p_hidepromo="${false}" />  
177                      <iselse>  
178                          <isdisplayliproduct p_productli="${lineItem}" p_formli="${FormLi}" p_editable="${true}" p_hideprice="${true}" p_hidepromo="${false}" />  
179                      </iselse>  
180                  </isif>  
181              </td>  
182              <isif condition="${dw.system.Site.getCurrent().getCustomPreferenceValue('enableStorePickUp')}">  
183                  <td class="item-delivery-options">  
184                      <isinclude template="checkout/cart/storepickup/deliveryoptions" />  
185                  </td>  
186              </isif>  
187              <td class="item-quantity">  
188                  <isif condition="${lineItem.bonusProductLineItem}">  
189                      <isprint value="${lineItem.quantity}" />  
190                  <iselse/>  
191                      <input type="number" min="0" name="${FormLi.quantity.htmlName}" maxlength="6" value="${lineItem.quantity.value.toFixed()}" class="input-t  
192                  </isif>  
193              </td>  
194          </tr>
```





4.10 Remove Add to Wishlist from cart for Extend products

- In: */app_storefront_core/cartridge/templates/default/checkout/cart/cart.isml
- Where: line 212

Replace:

```
<isif condition="${empty(pdct.ProductAddedToWishlist) || pdct.ProductAddedToWishlist != lineItem.productID}">
    <isif condition="${!isInWishList && !empty(lineItem.product)}">
        <a class="add-to-wishlist"
href="${URLUtils.https('Cart-AddToWishlist', 'pid', lineItem.productID)}"
name="${FormLi.addToWishList.htmlName}"
title="${Resource.msg('global.addthisitemtowishlist.title','locale',null)}">
            ${Resource.msg('global.addtowishlist','locale',null)}
        </a>
    </isif>
<iselse/>
    <div class="in-wishlist">
        ${Resource.msg('global.iteminwishlist','locale',null)}
    </div>
</isif>
```

With:

```
<isif condition="${!lineItem.custom.parentLineItemUUID}">
    <isif condition="${empty(pdct.ProductAddedToWishlist) || pdct.ProductAddedToWishlist != lineItem.productID}">
        <isif condition="${!isInWishList && !empty(lineItem.product)}">
            <a class="add-to-wishlist"
href="${URLUtils.https('Cart-AddToWishlist', 'pid', lineItem.productID)}"
name="${FormLi.addToWishList.htmlName}"
title="${Resource.msg('global.addthisitemtowishlist.title','locale',null)}">
                ${Resource.msg('global.addtowishlist','locale',null)}
            </a>
        </isif>
    <iselse/>
        <div class="in-wishlist">
            ${Resource.msg('global.iteminwishlist','locale',null)}
        </div>
    </isif>
</isif>
```





```
208 <div class="item-user-actions">
209     <isif condition="${!lineItem.bonusProductLineItem || lineItem.getBonusDiscountLineItem() != null}">
210         <button class="button-text" type="submit" value="${Resource.msg('global.remove','locale',null)}" name="${FormLi.deleteProduct.htmlName}"><span>${Resource.
211     </isif>
212     <isif condition="${!lineItem.custom.parentLineItemUUID}">
213         <isif condition="${!empty(pdct.ProductAddedToWishlist) || pdct.ProductAddedToWishlist != lineItem.productID}">
214             <a class="add-to-wishlist" href="${URLUtils_https('Cart-AddToWishlist', 'pid', lineItem.productID)}" name="${FormLi.addToWishList.htmlName}">
215                 ${Resource.msg('global.addtowishlist','locale',null)}
216             </a>
217         </isif>
218     </iselse/>
219     <div class="in-wishlist">
220         ${Resource.msg('global.iteminwishlist','locale',null)}
221     </div>
222     </isif>
223 </div>
224 </isif>
225 </div>
226
```

4.11 Add contract generation snippet to COPlaceOrder.js

- In: */app_storefront_controllers/cartridge/controllers/COPlaceOrder.js
- Where: function submit(), in the if (!orderPlacementStatus.error) statement

Code:

```
// Add Extend products to Contracts queue
var extendHelpers = require('*-/cartridge/scripts/extend');
addContractToQueue(order.object);
```

```
164     } else if (handlePaymentsResult.missingPaymentInfo) {
165         return Transaction.wrap(function () {
166             OrderMgr.failOrder(order);
167             return {
168                 error: true,
169                 PlaceOrderError: new Status(Status.ERROR, 'confirm.error.technical')
170             };
171         });
172     }
173
174     var orderPlacementStatus = Order.submit(order);
175     if (!orderPlacementStatus.error) {
176         clearForms();
177
178         // Add Extend products to Contracts queue
179         var extendHelpers = require('*-/cartridge/scripts/extendHelpers');
180         extendHelpers.addContractToQueue(order);
181     }
182
183 }
```





4.12 calculate.js hook

- Open your calculate.js hook script (*/cartridge/scripts/cart/calculate.js)
- Add the following script at the beginning of the script file

Code:

```
// Extend normalizationCartHook
var normalizeCartQuantities =
require('*/*cartridge/scripts/normalizationCartHook');
```

```
11  var HashMap = require('dw/util/HashMap');
12  var PromotionMgr = require('dw/campaign/PromotionMgr');
13  var ShippingMgr = require('dw/order/ShippingMgr');
14  var ShippingLocation = require('dw/order/ShippingLocation');
15  var TaxMgr = require('dw/order/TaxMgr');
16  var Logger = require('dw/system/Logger');
17  var Status = require('dw/system>Status');
18  var HookMgr = require('dw/system/HookMgr');
19  var collections = require('*/*cartridge/scripts/util/collections');
20
21 // Extend normalization cart hook
22 var normalizeCartQuantities = require('*/*cartridge/scripts/normalizationCartHook');
23
24 /**
25 * @function calculate
26 *
27 * calculate is the arching logic for computing the value of a basket. It makes
28 * calls into cart/calculate.js and enables both SG and OCAPI applications to share
29 * the same cart calculation logic.
```

- Find the function calculateProductPrices(basket) and add the following code afterwards. This snippet adds the normalizeCartQuantities(basket) function and an additional calculateProductPrices(basket) function to recalculate Extend Protection Plan prices after cart normalization. This snippet should be added before the tax calculation hook is called:

Code:

```
// =====
// ===== NORMALIZE CART QUANTITIES FOR EXTEND ITEMS =====
// =====
normalizeCartQuantities(basket);

// recalculate products prices after normalization
// quantities
calculateProductPrices(basket);
```





```
71
72     PromotionMgr.applyDiscounts(basket);
73
74     // since we might have bonus product line items, we need to
75     // reset product prices
76     calculateProductPrices(basket);
77
78     // =====
79     // =====  NORMALIZE CART QUANTITIES FOR EXTEND ITEMS  =====
80     // =====
81     normalizeCartQuantities(basket);
82
83
84     // recalculate products prices after normalization
85     // quantities
86     calculateProductPrices(basket);
87
88
89     // =====
90     // =====      CALCULATE TAX          =====
91     // =====
92     calculateTax(basket);
93
94
95     // =====
96     // =====      CALCULATE BASKET TOTALS    =====
97     // =====
98
99     basket.updateTotals();
100
101    // =====
102    // =====      DONE          =====
103    // =====
104
105    return new Status(Status.OK);
106
107
```

- Find the function calculateProductPrices(basket) {...}
- Change the “handle non-catalog products” if statement to be:

```
if (!productLineItem.catalogProduct ||  
productLineItem.custom.parentLineItemUUID)
```





```
96  /**
97  * @function calculateProductPrices
98  *
99  * Calculates product prices based on line item quantities. Set calculates prices
100 * on the product line items. This updates the basket and returns nothing
101 *
102 * @param {object} basket The basket containing the elements to be computed
103 */
104 function calculateProductPrices(basket) {
105     // get total quantities for all products contained in the basket
106     var productQuantities = basket.getProductQuantities();
107     var productQuantitiesIt = productQuantities.keySet().iterator();
108
109     // get product prices for the accumulated product quantities
110     var productPrices = new HashMap();
111
112     while (productQuantitiesIt.hasNext()) {
113         var prod = productQuantitiesIt.next();
114         var quantity = productQuantities.get(prod);
115         productPrices.put(prod, prod.priceModel.getPrice(quantity));
116     }
117
118     // iterate all product line items of the basket and set prices
119     var productLineItems = basket.getAllProductLineItems().iterator();
120     while (productLineItems.hasNext()) {
121         var productLineItem = productLineItems.next();
122
123         // handle non-catalog products
124         if (!productLineItem.catalogProduct || productLineItem.custom.parentLineItemUUID) {
125             productLineItem.setPriceValue(productLineItem.basePrice.valueOrDefault());
126             continue;
127         }
128
129         var product = productLineItem.product;
130
131         // handle option line items
132         if (productLineItem.optionProductLineItem) {
133             // for bonus option line items, we do not update the price
134             // the price is set to 0.0 by the promotion engine
135             if (!productLineItem.bonusProductLineItem) {
```

4.13 Include extend_header.isml

- In:
*/app_storefront_core/cartridge/templates/default/components/header/htmlhead_Ul.isml
- Where: at the end of the file

Code:

```
<iscomment>Extend Integration</iscomment>
<isinclude template="components/footer/extend_header"/>
```





```
1 <iscontent type="text/html" charset="UTF-8" compact="true"/>
2 <iscomment>THIS FILE ONLY CONTAINS DIFFERENCES BETWEEN SIMPLE AND RICH UI LIB INCLUDES</iscomment>
3 <iscomment>
4 The <!--BEGIN/END--> comments are control statements for the build cartridge which can be found in xChange https://xchange.demandware.com/doc
5 If you are not using the build cartridge the comments can be safely removed.
6 </iscomment>
7
8 <iscomment>THIRD PARTY STYLESHEETS STYLING</iscomment>
9 <link href="${URLUtils.staticURL('/lib/jquery/ui/jquery-ui.min.css')}" type="text/css" rel="stylesheet" />
10 <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css" rel="stylesheet">
11
12 <iscomment>Extend Integration</iscomment>
13 <isinclude template="components/header/extend_header"/>
14
15 <!--[if lt IE 9]>
16 <script src="${URLUtils.staticURL('/js/lib/html5.js')}"></script>
17 <![endif]-->
18
19 <iscomment>Salesforce Commerce Cloud active data scripts</iscomment>
20 <isactivedatahead/>
21 ~
```

4.14 Add data attribute to link in minilineitems.isml

- In:
*/app_storefront_core/cartridge/templates/default/checkout/components/minilineitems.isml
- Where: line 89

Code:

```
<a href="${itemUrl}"
data-is-extend-product="${productLineItem.custom.parentLineItemUUID ? true : false}" title="${Resource.msgf('product.label','product',null,
productLineItem.productName)}">
<isprint value="${productLineItem.productName}" />
</a>
```

4.15 CartView.js

- In:
/cartridges/app_storefront_controllers/cartridge/scripts/views/CartView.js
- Where: line 38
- Move the following code snippet from line 38 to line 33 to ensure the warranty price is recalculated before displaying in the cart.

Code:

```
// Refreshes the cart calculation
Transaction.wrap( function () {
```





```
    Cart.get(cart).calculate();
})
```

See screenshots below:

Code location **before**:

```
var CartView = View.extend({

  /**
   * Updates shipments, coupons, and cart calculation for the view and validates
   * for checkout.
   */
  prepareView: function () {

    var cart = this.Basket;
    if (cart) {

      // Refreshes shipments.
      session.forms.cart.shipments.copyFrom(cart.shipments);
      // Refreshes coupons.
      session.forms.cart.coupons.copyFrom(cart.couponLineItems);

      // Refreshes the cart calculation.
      Transaction.wrap(function () {
        Cart.get(cart).calculate();
      });

      var validationResult = Cart.get(cart).validateForCheckout();
      this.EnableCheckout = validationResult.EnableCheckout;
      this.BasketStatus = validationResult.BasketStatus;
      this.WishList = customer.authenticated ? require('~/cartridge/scripts/r

      // Extend Analytics.
      extendAnalyticsHelpers.setUpdateCartPayload(cart);
      this.analyticsPayload = extendAnalyticsHelpers.getAnalyticsPayload();
    }

    return;
  },
});
```





Code location **after**:

```
1  var CartView = View.extend([
2
3
4    /**
5     * Updates shipments, coupons, and cart calculation for the view and validates the
6     * for checkout.
7     */
8    prepareView: function () {
9
10      var cart = this.Basket;
11      if (cart) {
12
13        // Refreshes the cart calculation.
14        Transaction.wrap(function () {
15          |   Cart.get(cart).calculate();
16        });
17
18        // Refreshes shipments.
19        session.forms.cart.shipments.copyFrom(cart.shipments);
20        // Refreshes coupons.
21        session.forms.cart.coupons.copyFrom(cart.couponLineItems);
22
23        var validationResult = Cart.get(cart).validateForCheckout();
24        this.EnableCheckout = validationResult.EnableCheckout;
25        this.BasketStatus = validationResult.BasketStatus;
26        this.WishList = customer.authenticated ? require('~/cartridge/scripts/model'
27
28        // Extend Analytics.
29        extendAnalyticsHelpers.setUpdateCartPayload(cart);
30        this.analyticsPayload = extendAnalyticsHelpers.getAnalyticsPayload();
31      }
32
33      return;
34    },
35  ],
```





4.15 include ExtendAnalyticsHelpers in CartView.js

- In:
/cartridges/app_storefront_controllers/cartridge/scripts/views/CartView.js
- Where: top of the file

Code:

```
var extendAnalyticsHelpers =
require('*cartridge/scripts/extendAnalyticsHelpers')
```

```
/**
 * View used to render the cart. This view makes sure the coupons, shipments, and basket
 * calculation are up to date before rendering the cart.
 * @module views/CartView
 */
var View = require('./View');

var Cart = require('~/cartridge/scripts/models/CartModel');
var Transaction = require('dw/system/Transaction');
var extendAnalyticsHelpers = require('*cartridge/scripts/extendAnalyticsHelpers')
```

- In:
/cartridges/app_storefront_controllers/cartridge/scripts/views/CartView.js
- Where: “prepareView” function → line 48

Code:

```
// Extend Analytics
extendAnalyticsHelpers.setUpdateCartPayload(cart)
this.analyticsPayload = extendAnalyticsHelpers.getAnalyticsPayload()
```





```
22 var CartView = View.extend([
23
24     /**
25      * Updates shipments, coupons, and cart calculation for the view and validates the
26      * for checkout.
27      */
28     prepareView: function () {
29
30         var cart = this.Basket;
31         if (cart) {
32
33             // Refreshes the cart calculation.
34             Transaction.wrap(function () {
35                 Cart.get(cart).calculate();
36             });
37
38             // Refreshes shipments.
39             session.forms.cart.shipments.copyFrom(cart.shipments);
40             // Refreshes coupons.
41             session.forms.cart.coupons.copyFrom(cart.couponLineItems);
42
43             var validationResult = Cart.get(cart).validateForCheckout();
44             this.EnableCheckout = validationResult.EnableCheckout;
45             this.BasketStatus = validationResult.BasketStatus;
46             this.WishList = customer.authenticated ? require('~/cartridge/scripts/model'
47
48             // Extend Analytics.
49             extendAnalyticsHelpers.setUpdateCartPayload(cart);
50             this.analyticsPayload = extendAnalyticsHelpers.getAnalyticsPayload();
51         }
52
53         return;
54     },
55 ])
```





4.16 include ExtendAnalyticsHelpers.setDeleteProductPayload in Cart.js

- In:
/cartridges/app_storefront_controllers/cartridge/controllers/Cart.js
- Where: top of the file

Code:

```
var extendHelpers = require('*cartridge/scripts/extendHelpers');
var extendAnalyticsHelpers =
require('*cartridge/scripts/extendAnalyticsHelpers');
```

```
1  'use strict';
2
3 /**
4  * Controller that adds and removes products and coupons in the cart.
5  * Also provides functions for the continue shopping button and minicart.
6  *
7  * @module controllers/Cart
8 */
9
10 /* API Includes */
11 var ArrayList = require('dw/util/ArrayList');
12 var ISML = require('dw/template/ISML');
13 var Resource = require('dw/web/Resource');
14 var Transaction = require('dw/system/Transaction');
15 var URLUtils = require('dw/web/URLUtils');
16
17 /* Script Modules */
18 var app = require('~/cartridge/scripts/app');
19 var guard = require('~/cartridge/scripts/guard');
20
21 var extendHelpers = require('*cartridge/scripts/extendHelpers')
22 var extendAnalyticsHelpers = require('*cartridge/scripts/extendAnalyticsHelpers')
```

- In:
/cartridges/app_storefront_controllers/cartridge/controllers/Cart.js
- Where: “submitForm” function → “deleteProduct” function → line 182

Code:

```
var warrantyLineItem = extendHelpers.checkForWarrantyLI(cart,
formgroup.getTriggeredAction().object);
extendAnalyticsHelpers.setDeleteProductPayload(cart,
formgroup.getTriggeredAction().object, warrantyLineItem);
```





```
177     };
178 },
179 'deleteProduct': function (formgroup) {
180     // EXTEND - Check if there is a warranty associated with the product that is being deleted
181     var warrantyLineItem = extendHelpers.checkForWarrantyLI(cart, formgroup.getTriggeredAction().object);
182     extendAnalyticsHelpers.setDeleteProductPayload(cart, formgroup.getTriggeredAction().object, warrantyLineItem);
183     Transaction.wrap(function () {
184         cart.removeProductLineItem(formgroup.getTriggeredAction().object);
185
186         // EXTEND - if there is a warranty associated with the product remove the corresponding warranty
187         if (warrantyLineItem) {
188             cart.removeProductLineItem(warrantyLineItem);
189         }
190     });
191
192     return [
193         cart
194     ];
195 },
```

4.17 include “Extend Integration” and “Extend Analytics Integration” in app.js

- In:
/cartridges/app_storefront_core/cartridge/static/default/js/app.js
- Where:
“addToCart” function. Line 2537 - 2547

Code:

```
// Extend Integration
if ($('#extend-offer').length || (window.EXT_GLOBAL_SWITCH &&
window.EXT_PDP_UPSELL_SWITCH)) {
    extendAddToCart($form, page, minicart, dialog, addItemToCart);
    return;
}
// ExtendAnalytics Integration
if (window.EXT_A_EXT_GLOBAL_SWITCH) {
    var payload = createAddToCartAnalytics($form)
    trackAddToCart(payload)
}
```





```
2529
2530 /**
2531 * @description Handler to handle the add to cart event
2532 */
2533 var addToCart = function (e) {
2534     e.preventDefault();
2535     var $form = $(this).closest('form');

2536
2537     // Extend Integration
2538     if ($('#extend-offer').length || (window.EXT_GLOBAL_SWITCH && window.EXT_PDP_UPSELL_SWITCH)) {
2539         extendAddToCart($form, page, minicart, dialog, addItemToCart);
2540         return;
2541     }

2542
2543     // ExtendAnalytics Integration
2544     if (window.EXT_A_EXT_GLOBAL_SWITCH) {
2545         var payload = createAddToCartAnalytics($form)
2546         trackAddToCart(payload)
2547     }

2548
2549     addItemToCart($form).then(function (response) {
2550
```

5. External Interfaces

All requests are done through the Extend API. The full reference guide, along with the resource structure for requests, can be found on the developer site - <https://developers.extend.com/latest>.

6. Firewall Requirements

None





7. Orders API

7.1. Custom preferences

- i. Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → Extend
- ii. Configure the provided Extend credentials in the following fields (refer to Section 2.2 for additional info):
 - Access Token
 - Store ID
- iii. Configure the following Orders API specific preferences:
 - **Extend API Version:** set this to the latest Extend API version.
 - **Extend API Method:** set this to Orders API (please confirm with the Extend project team before changing versions).

7.2. Testing

i. **LeadToken:**

Go to PDP → Choose product variation → Click “Add to Cart” button without an Extend Protection Plan (LeadTokens are only generated for warrantable products if the customer did not elect to protect their purchase originally) → Go to cart → Go to checkout page → Make a purchase.

Result:

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Shipment → Choose ProductLineItem → Attributes





General

Attributes

Attributes for "Black Flat Front Wool Suit"

On this page you can edit the attributes of the product line item. Fields with a red asterisk (*) are mandatory. Click **Apply** to save changes.

Extend

Extend Contract ID: No data is available

Extend Refund Statuses:

Lead Token: f102f17d9fda6fe8cd8a76962ae68dd9

ii. Extend Contract ID:

Go to PDP → Choose product variation → Choose an extension plan → Click “Add to Cart” button with an Extend Protection Plan → Go to cart → Go to checkout page → Make a purchase.

Result:

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Shipment → Choose ExtendLineItem → Attributes





General Attributes

Attributes for "Extend Product Protection: 2 years"

On this page you can edit the attributes of the product line item. Fields with a red asterisk (*) are mandatory. Click **Apply** to save changes.

Extend

Extend Contract ID: 764a42a5-c9f2-442d-9916-03061b22bf97

Extend Refund Statuses:

Lead Token:

iii. Refund status:

- Go to Merchant Tools → Ordering → Orders → Choose an Order with at least 1 created Extend Contract ID → Order Status → Choose “Cancelled” → Click “Apply” button.
- Then Go To Administration → Operations → Jobs → “Create Refund from SFCC” → Click “Run Now” button.

Result:

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Shipment → Choose ExtendLineItem → Attributes:





Merchant Tools > Ordering > Orders > Order: 00006006(SiteGenesis) > Shipment: 00032506 > Line Item: EXTEND-24

General

Attributes

Attributes for "Extend Product Protection: 2 y"

On this page you can edit the attributes of the product line item. Fields with a red asterisk (*) are mandatory. Click **Apply** to save changes.

Extend

Extend Contract ID: 764a42a5-c9f2-442d-9916-03061b22bf97

Extend Refund Statuses: {"764a42a5-c9f2-442d-9916-03061b22bf97": "refund_paid"}

Lead Token:

Go to Merchant Tools → Ordering → Orders → Choose Your Order → Attributes:





[Merchant Tools](#) > [Ordering](#) > [Orders](#) > Order: 00006006(SiteGenesis)

General

Attributes

Payment

Notes

History

Attributes for Order '00006006'

On this page you can edit the attributes of the order. Fields with a red asterisk (*) are mandatory. Click

Extend

Extend Refund Status:

Refund Statuses:

refund_quoted: Refund amounts have been previewed by merchant/customer at least once, but merchant has not reported that they have refunded the customer yet.

refund_paid: Refund has been reported as paid by the merchant. Contract is canceled. Finance will credit the merchant in the next monthly invoice.

refund_denied: Customer is not eligible for a refund. Customer has consumed all entitlements associated with the contract. Get Refund via Contract ID.

ERROR: service call error.





8. Extend SDK analytics

8.1. Custom Preferences

- i. Go to Business Manager → Merchant Tools → Site Preferences → Custom Preferences → Extend
- ii. Configure the provided Extend credentials in the following fields (refer to Section 2.2 for additional info):
 - Access Token
 - Store ID
- iii. Ensure “Extend Analytics Switch” is set to “Yes”.

Extend Analytics Switch	Yes
(extendAnalyticsSwitch)	

8.2. Tracking events

i. Extend.trackOfferViewed:

This method is triggered when an Extend warranty offer is rendered on the screen to a user.

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> Product-Productnav?lang=en_US&start=1&pid=25686514&format/ajax <input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686... <input type="checkbox"/> Product-Variation?pid=25686514&dwvar_25686514_size...WL&Quantity=1&... <input type="checkbox"/> offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686... <input type="checkbox"/> Product-Variation?pid=25686514&dwvar_25686514_widt...WL&Quantity=1... <input type="checkbox"/> tracking		<p>Request Payload view source</p> <pre>▼ {type: "analytics",...} ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_viewed",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" ▶ context: {,...} eventName: "offer_viewed" platform: "extend-sdk-client" properties: {productId: "750518548203", offerType: {area: "product_page", component: "buttons"}} ▶ offerType: {area: "product_page", component: "buttons"} productId: "750518548203" sessionState: {products: [], productsWithWarranties: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643290228798 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				

ii. Extend.trackProductAddedToCart:

This method is triggered when an user adds a (non-Extend) product to the cart. This event accepts the product's “productId” and the number of units (quantity) added to the cart.





Name	x	Headers	Payload	Preview	Response	Initiator	Timing
<pre>https://api-demo.helloextend.com/tracking?_t=1&pid=25686514&format=ajax □ offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686... □ Product-Variation?pid=25686514&dwvar_25686514_size...WL&Quantity=1&... □ offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=25686... □ Product-Variation?pid=25686514&dwvar_25686514_widt...WL&Quantity=1... □ tracking □ offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=75051... □ tracking □ tracking □ Cart-AddProduct?format=ajax</pre>	x		<pre>▼ Request Payload view source ▷ {type: "analytics",...} ▷ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "product_added_to_cart",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" context: {...} eventName: "product_added_to_cart" platform: "extend-sdk-client" ▷ properties: {productId: "750518548203", productQuantity: 1} productId: "750518548203" productQuantity: 1 ▷ sessionState: {products: [{productId: "750518548203", productQuantity: 1}], productsWithWarranties: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643290587268 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				

iii. Extend.trackOfferAddedToCart

This method is triggered when an user adds an Extend warranty to the cart. This event accepts the product's "productId", the number of units (quantity), and the Extend warranty's "planId".

Name	x	Headers	Payload	Preview	Response	Initiator	Timing
<pre>□ tracking □ Extend-AddExtendProduct □ tracking □ Extend-IsEligibleForWarranty?uuid=7896859ab60bb9ab25a226641f □ Extend-IsEligibleForWarranty?uuid=d35210df93d0915974167902d1</pre>	x		<pre>▼ Request Payload view source ▷ {type: "analytics",...} ▷ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_added_to_cart",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" context: {...} eventName: "offer_added_to_cart" platform: "extend-sdk-client" ▷ properties: {productId: "750518548203", productQuantity: 1, warrantyQuantity: 1,...} offerType: {area: "cart_page", component: "modal"} planId: "10001-misc-elec-base-replace-3y" productId: "750518548203" productQuantity: 1 warrantyQuantity: 1 ▷ sessionState: {products: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643290855612 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				

iv. Extend.trackOfferRemovedFromCart

This method is triggered when an user removes an Extend warranty from the cart. This event accepts the Extend warranty's "planId" and the associated "productId" of the product the warranty would have covered.

Name	x	Headers	Payload	Preview	Response	Initiator	Timing
<pre>https://api-demo.helloextend.com/tracking □ Extend-IsEligibleForWarranty?uuid=896859ab60bb9ab25a226641f □ tracking □ offers?storeId=5bcd9cf9-807a-4389-ba29-6d38daf7d395&productId=75051... □ tracking</pre>	x		<pre>▼ Request Payload view source ▷ {type: "analytics",...} ▷ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_removed_from_cart",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" context: {...} eventName: "offer_removed_from_cart" platform: "extend-sdk-client" ▷ properties: {productId: "750518548203", planId: "10001-misc-elec-base-replace-3y" planId: "10001-misc-elec-base-replace-3y" productId: "750518548203" ▷ sessionState: {products: [{productId: "750518548203", productQuantity: 1}], productsWithWarranties: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643291063318 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				





v. Extend.trackOfferUpdated

This method is triggered when an user increments or decrements the quantity of a warranty that has already been added to the cart. This event takes the Extend warranty's "planId", the associated "productId" of the product the warranty covers, as well as an update object containing the set of updates to apply to the warranty offer. **If the quantity of warranty is updated to 0, the a Extend.trackOfferRemoved event is called**, and further updates to this planId/productId will result in a no-op until it is re-added via Extend/trackOfferAddedToCart.

Name	x Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> Extend.IsEligibleForWarranty?uuid=7896859ab60bb9ab25a226641f <input type="checkbox"/> Extend.IsEligibleForWarranty?uuid=eeeab5379a83081e350fc5f3aa <input checked="" type="checkbox"/> tracking		<pre>▼ Request Payload view source ▼ {type: "analytics",...} ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_updated",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" ▶ context: {,} eventName: "offer_updated" platform: "extend-sdk-client" properties: {productId: "750518548203", planId: "10001-misc-elec-base-replace-ly",...} ▶ newQuantities: {warrantyQuantity: 2, productQuantity: 2} ▶ oldQuantities: {productQuantity: 1, warrantyQuantity: 1} planId: "10001-misc-elec-base-replace-ly" productId: "750518548203" ▶ sessionState: {products: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643291432640 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				

vi. Extend.trackProductRemovedFromCart

This method is triggered when an user removes a product from the cart that **does not** have a warranty offer associated with it. This event takes the "productId" of the product being removed from the cart.

Name	x Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> tracking		<pre>▼ Request Payload view source ▼ {type: "analytics",...} ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "product_removed",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" ▶ context: {,} eventName: "product_removed" platform: "extend-sdk-client" properties: {productId: "750518548203"} productId: "750518548203" ▶ sessionState: {products: [], productsWithWarranties: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643291664323 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				

vii. Extend.trackProductUpdated

This method is triggered when an user increments or decrements the quantity of a product that has already been added to the cart. This event takes the "productId" of the product being updated, as well as an update object





containing the set of updates to apply to the product. The product being updated must **not** be associated with a warranty offer.

If the “productQuantity” passed into this method is 0, then Extend.trackProductRemoveFromCart is fired.

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> Extend-IsEligibleForWarranty?uuid=f48cc344ec25b9d94f038586e8 <input type="checkbox"/> tracking <input type="checkbox"/> tracking		<pre>▼ Request Payload view source ▼ {type: "analytics",...} ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "product_updated",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" ▶ context: {...} eventName: "product_updated" platform: "extend-sdk-client" ▼ properties: {productId: "750518699608M", oldProductQuantity: 1, newProductQuantity: 2} newProductQuantity: 2 oldProductQuantity: 1 productId: "750518699608M" ▶ sessionState: {products: [{productId: "750518699608M", productQuantity: 2}], productsWithWarranties: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643292010508 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				

viii. Extend.trackCartCheckout

This method is triggered when a customer completes a purchase.

- After the Checkout flow is complete (at Order Confirmation page), there fires two tracking events - offer_sold, and total_cart_revenue

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> COShipping-GetApplicableShippingMethods.JSON?address...ess2=&country... <input type="checkbox"/> COShipping-GetApplicableShippingMethods.JSON?address...ountryCode=u... <input type="checkbox"/> COShipping-GetApplicableShippingMethods.JSON?address...s&stateCode=... <input type="checkbox"/> COBilling-SelectCreditCard?creditCardUUID=3a1e6870902235f9c0771e4bf1 <input type="checkbox"/> tracking <input type="checkbox"/> tracking		<pre>▼ Request Payload view source ▼ {type: "analytics",...} ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "offer_sold",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" ▶ context: {...} eventName: "offer_sold" platform: "extend-sdk-client" ▼ properties: {offerType: {area: "product_page", component: "buttons"}, planId: "10001-misc-elec-base-replace-3y",...} offerType: {area: "product_page", component: "buttons"} planId: "10001-misc-elec-base-replace-3y" productId: "75051894485" productQuantity: 1 warrantyQuantity: 1 ▶ sessionState: {products: [], productsWithWarranties: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643292435532 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				

Name	Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> COShipping-GetApplicableShippingMethods.JSON?address...ess2=&country... <input type="checkbox"/> COShipping-GetApplicableShippingMethods.JSON?address...ountryCode=u... <input type="checkbox"/> COShipping-GetApplicableShippingMethods.JSON?address...s&stateCode=... <input type="checkbox"/> COBilling-SelectCreditCard?creditCardUUID=3a1e6870902235f9c0771e4bf1 <input type="checkbox"/> tracking <input type="checkbox"/> tracking		<pre>▼ Request Payload view source ▼ {type: "analytics",...} ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "total_cart_revenue",...} cartId: "0BF2F4B4-7570-4A4E-99E5-416E0840CA88" ▶ context: {...} eventName: "total_cart_revenue" platform: "extend-sdk-client" ▼ properties: {cartTotal: 409.47} cartTotal: 409.47 ▶ sessionState: {products: [], productsWithWarranties: [],...} storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395" timestamp: 1643292435534 userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7" type: "analytics"</pre>				





- After the Checkout flow is complete (at Order Confirmation page), there fires two tracking events called - product_sold and total_cart_revenue

Name	x Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> tracking						

Request Payload view source

```
▼ {type: "analytics",...}
  ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "product_sold",...}
    cartId: "669FA4D1-70F4-4463-B816-28855403B9C5"
    ▶ context: {,...}
      eventName: "product_sold"
      platform: "extend-sdk-client"
    ▼ properties: {productId: "750518894485", productQuantity: 1}
      productId: "750518894485"
      productQuantity: 1
    ▶ sessionState: {products: [{productId: "750518894485", productQuantity: 1}], productsWithWarranties: [],...}
      storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"
      timestamp: 164329273392
      userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7"
      type: "analytics"
```

Name	x Headers	Payload	Preview	Response	Initiator	Timing
<input type="checkbox"/> tracking						

Request Payload view source

```
▼ {type: "analytics",...}
  ▼ object: {userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7", eventName: "total_cart_revenue",...}
    cartId: "669FA4D1-70F4-4463-B816-28855403B9C5"
    ▶ context: {,...}
      eventName: "total_cart_revenue"
      platform: "extend-sdk-client"
    ▼ properties: {cartTotal: 325.48}
      cartTotal: 325.48
    ▶ sessionState: {products: [], productsWithWarranties: [],...}
      storeId: "5bcd9cf9-807a-4389-ba29-6d38daf7d395"
      timestamp: 164329273393
      userId: "1B5D4861-6DE1-4552-96E6-65624D3E8EB7"
      type: "analytics"
```





Testing

1. Enable Extend

- Ensure the following switches are set to Yes
 - Enable Extend
 - Enable in PDP
 - Enable Up-sell in PDP
 - Enable Up-sell in Cart
- Ensure the product catalog has been exported to Extend and confirm with the Extend project team that warranty plans are enabled in the system.

2. PDP

2.1. Extend Plan

Navigate to any simple or variant product that has an Extend Protection Plan enabled. Select a variation and confirm an Extend offer is shown.

The screenshot shows a product detail page for a "Must Have Washable No-Iron Georgette Blouse". The product is a white blouse with a collar and buttons. The page includes a search bar, navigation links for NEW ARRIVALS, WOMENS, MENS, ELECTRONICS, and TOP SELLERS, and a shopping cart icon showing 6 items. The product details section shows the item number 00885538465M, the price of \$69.00, and options to select color (Black, White, Ivory) and size (6, 8, 10, 12). A red box highlights the "Add product protection from Extend" button, which offers three options: 1 Year - \$6.99, 2 Years - \$12.99, and 3 Years - \$17.99. Below the button are social sharing icons for Facebook, Twitter, Google+, Pinterest, and Email, along with "Add to Wishlist" and "Add to gift registry" buttons.





2.2. Add to Cart

Select an Extend Protection Plan and click Add to Cart. Confirm 2 products were added to the cart – the SFCC catalog product along with the Extend Protection Plan.

2.3. Up-sell Modal

Navigate to any simple or variant product that has an Extend Protection Plan enabled and click Add to Cart on the PDP without selecting any Extend Protection Plan. Confirm the Extend Interstitial Modal is presented. Selecting “Add Protection” will add the Extend Protection Plan to the cart, along with the catalog product. Selecting “No, thanks” will only add the catalog product to the cart.

The screenshot shows a product detail page for a "Must Have Washable No-Iron Georgette Blouse" with item number 008885538458. The price is \$69.00. A "SELECT COLOR" section shows options for Black, Green, and Ivory. Below the product image, a modal window from Extend is displayed. The modal header features the Extend logo and a small profile picture of a user. The text reads: "Protect your purchase from drops, breaks, spills, and defects with Extend." Underneath, it says "This plan covers:" followed by a bulleted list: "Fast and free product replacements", "Accidental damage such as breaks, drops, and spills", and "Extended malfunction and wear-and-tear protection". There are three plan options: "1 Year Protection Plan \$6.99", "2 Year Protection Plan \$11.99" (marked as "Best Seller"), and "3 Year Protection Plan \$17.99" (marked as "Best Value"). At the bottom of the modal are two buttons: "No, thanks" and a red "Add protection" button. To the right of the modal, a large image of a smiling woman holding a package is visible.





3. Cart

3.1. Extend Plans in Cart

Confirm an Extend plan has been added to the cart.

FREE 2-Day SHIPPING FOR ORDERS OVER \$300

Enter Keyword or Item No.

NEW ARRIVALS WOMENS MENS ELECTRONICS TOP SELLERS

[« Continue Shopping](#) [CHECKOUT](#)

PRODUCT	DELIVERY OPTIONS	QTY	PRICE	TOTAL
Must Have Washable No-Iron Georgette Blouse Item No.: 008885538465M Color Ivory Size 8 Edit Details	Home Delivery	<input type="text" value="1"/>	In Stock \$69.00 Remove Add to Wishlist	\$69.00
 Extend Product Protection: 2 years for Must Have Washable No-Iron Georgette Blouse Item No.: EXTEND-24 Extend Term: 2Y	Home Delivery	<input type="text" value="1"/>	In Stock \$12.99 Remove	\$12.99

Enter coupon code

Subtotal \$81.99
Shipping -
Sales Tax -
Estimated Total \$81.99

[« Continue Shopping](#) [CHECKOUT](#)

3.2. Up-sell button for eligible products in cart

Add an eligible catalog product to the cart, without selecting any Extend Protection Plan for it. Navigate to the cart page and confirm the cart offer button is presented.



PRODUCT	DELIVERY OPTIONS	QTY	PRICE	TOTAL
 <p>Must Have Washable No-Iron Georgette Blouse Item No.: 008885538465M Color Ivory Size 8 Edit Details</p> <p>Add product protection from \$6.99</p>	Home Delivery	1	In Stock Remove Add to Wishlist	\$69.00
<input type="text" value="Enter coupon code"/> Apply Update Cart			Subtotal Shipping	\$69.00

3.3. Up-sell modal in cart

Click on the up-sell cart offer button and confirm the Extend Interstitial Modal is presented. Selecting “Add Protection” will add the corresponding Extend Protection Plan to the cart. Selecting “No, thanks” will close the modal.



Must Have Washable No-Iron Georgette Blouse
Item No.: 008885538458
Color Ivory
Size 6
[Edit Details](#)

[Add accident protection for \\$6.99](#)

Enter coupon code [Apply](#) [Update Cart](#)

 + 

Protect your purchase from drops, breaks, spills, and defects with Extend.

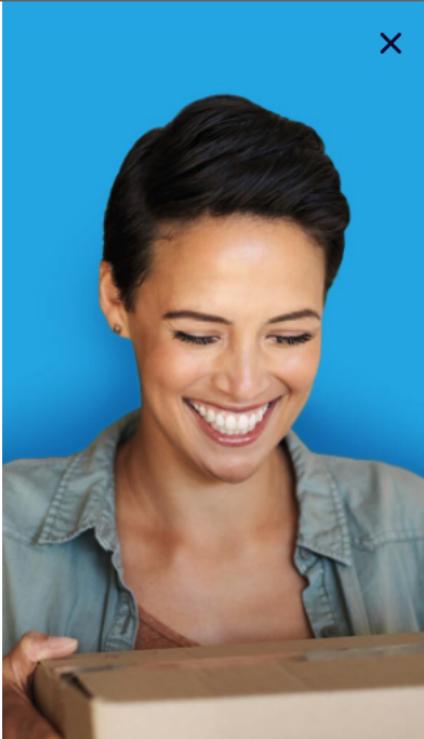
This plan covers:

- Fast and free product replacements
- Accidental damage such as breaks, drops, and spills
- Extended malfunction and wear-and-tear protection

[See plan details](#)  Offered and sold by Extend

1 Year Protection Plan \$6.99	2 Year Protection Plan \$11.99 <small>Best Seller!</small>	3 Year Protection Plan \$17.99 <small>Best Value!</small>
--	--	---

[No, thanks](#) [Add protection](#)



FEATURED PRODUCTS





4. Business Manager

4.1. ExtendContractsQueue custom object

After placing an order containing at least 1 Extend Protection Plan, a new instance of this custom object is created. This holds all the data needed for generating an Extend contract.

[Merchant Tools](#) > [Custom Objects](#) > [Custom Objects](#) > 8f6c7fc79e60c41637321dab6a - General

General

Manage '8f6c7fc79e60c41637321dab6a' (ExtendContractsQueue)

Fields with a red asterisk (*) are mandatory. You can view and edit the name and description in other languages, if required. Click **Apply** to save the details.

contractData	
Site Currency:	USD
Extend Warranty Plan:	{"purchasePrice":1364,"planId":"10001-misc-elec-base"}
SFCC Product:	{"referenceId":"008885538465M","purchasePrice":724}
Last Modified: *	12/18/2019 : 9:50 pm MM/dd/yyyy h:mm a
Creation Date: *	12/18/2019 : 9:50 pm MM/dd/yyyy h:mm a
Order Customer:	{"phone":"4157477799","email":"kris@extend.com","name":"Kris"}
Log:	(Empty)
Line Item UUID: *	8f6c7fc79e60c41637321dab6a
Order Total:	9,238.00 (Number)
Order Number:	00000907

4.2. Extend Contracts Creation Job

This job will process the ExtendContractsQueue object instances and will generate a contract for each instance. After successfully executing the request, the objects will be deleted.





Questions? Feedback?

At Extend, customer experience is our North Star, and that goes for both our merchant customers (you!) and your store's customers as they buy your products and our protection plans. So, we want to do anything we can to make your Extend integration successful and to make your customers' experiences outstanding.

If you have any questions, please don't hesitate to reach out to us any time or contact integrations@extend.com for help. And if you have any feedback about our documentation, integration process, or our APIs and SDK, we are all ears!

