

Fanplayr CommerceCloud Integration

Summary

Fanplayr tracks user behavior on your site, and allows you to target these users with offers and calls to action.

Both general page views and order details are tracked by Fanplayr.

You also have the option to add “Session Coupon” functionality to your site. This allows you to surface offers to your users, through Fanplayr, that will be active only to that user. This functionality makes sure that offers do not end up on offer sharing sites.

In this document we detail installation of the Fanplayr cartridge and the points at which it can be integrated into the SiteGenesis store. Note your store integration may vary.

Before you start you will need to get your ACCOUNT KEY from your Fanplayr representative.

Installation

To install our cartridge and integrate into your store you will go through the following steps:

- Import site preferences
- Import the Cartridge into UX Studio
- Add to Effective Cartridge Path
- Set up site preferences
- Add Fanplayr tag to your template
- Test tracking
- Test "Apply to Cart"
- Set up Session Offers

- Using Pipeline
- Using Controller
- Testing Session Offers

Import site preferences

The first thing to do is import the preferences

- Log into your Business Manager
- Go to Administration / Site Development / Import & Export
- Click Upload under Import & Export File

Import & Export Files

[Upload](#) and [download](#) your import and export files.

- Choose the file at `_/metadata/fanplayr_preferences.xml` and click Upload
- Go back to *Administration / Site Development / Import & Export* and click the Import link

Meta Data

[Import](#) and [export](#) your system meta data (i.e., system type extension)

- Select the newly uploaded file. Click Next >>.

Select	Name	File Size	Last Modified
	fanplayr_preferences.xml	3.24 KB	8/12/16 6:03:21 pm
			Next >> Cancel

- Step 3 should display the following information. Click the Import link at the bottom right.

Validation Status	
Current Action	Finished XML schema validation.
Validation Errors	0
Validation Warnings	0
Import File Content	
System Type Extensions	1
Custom Type Definitions	0
Attribute Definitions	6
Attribute Value Definitions	0
Attribute Group Definitions	1
Custom Preference Groups	0
Custom Preference Definitions	0

You should now have access to Fanplayr preferences at *Merchant Tools / Site Preferences / Custom Site Preferences*

Import Cartridge into UX Studio

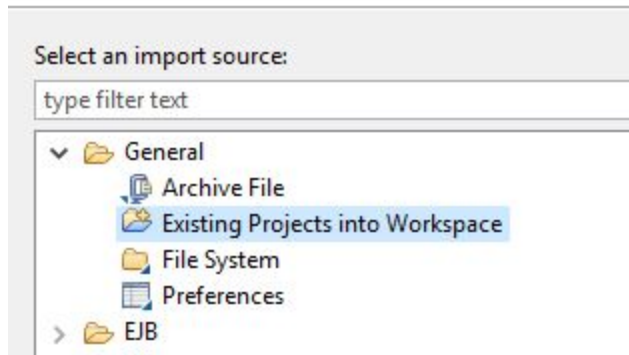
If you are using UX Studio to work with your site these steps show you how to import our Cartridge including Scripts, Controllers and Pipelines. If developing your site using another technique please contact us for help or follow the general rules for adding Cartridges there.

- Open UX Studio
- Go to the File / Import menu
- Choose General / Existing Projects into Workspace and click Next

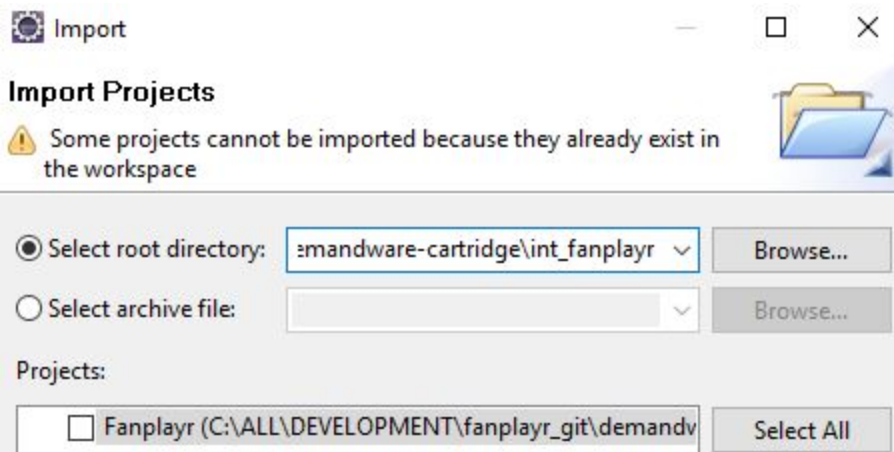


Select

Create new projects from an archive file or directory.



- Select the directory /int_fanplayr/ and then tick the box next to the Fanplayr listing. Click Finish. This should add the Fanplayr components to your project. You may be asked if you want to sync this with your store, please do that.



Add to Effective Cartridge Path

We now need to add Fanplayr to our Effective Cartridge Path. This let's your site know to run the code imported with Fanplayr on your site.

- Log in to your Business Manager
- Go to Administration / Sites / Manage Sites

- Choose your site
- Go to Settings
- Add the string “int_fanplayr:” (no quotes, make sure there is a colon at the end) to the start of Cartridges field.

General

Settings

Cache

Site Status

Fanplayr Demandware Store - Settings

Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

Instance Type: Sandbox/Development ▾

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new configuration and are intended only to support an older configuration style.

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges: int_fanplayr:app_storefront_controllers:ap

Effective Cartridge Path: int_fanplayr:app_storefront_controllers:app_sto

Fanplayr code should now be running on your site. You can make sure that it is running by going to /Fanplayr-Apply/ URL. This should take you to your cart if it is working, or an error page if it is not.

Set up site preferences

Before we get Fanplayr to start tracking we will need to add your Fanplayr ACCOUNT KEY to the preferences. If you do not have an ACCOUNT KEY, please contact your Fanplayr representative.

- Log in to your Business Manager
- Go to Merchant Tools / Site Preferences / Custom Preferences

Site Preferences

[Locking](#)
[Baskets](#)
[A/B Tests](#)
[Locales](#)
[Currencies](#)
[Source Codes](#)
[Gift Certificates](#)
[Guided Search](#)
[Sequence Numbers](#)
[Order](#)
[Coupons](#)
[Promotions](#)
[Storefront Toolkit](#)
[Storefront URLs](#)
[Custom Preferences](#)

- Click Fanplayr

ID	Name
Fanplayr	Fanplayr

The first two values, “Enable Fanplayr?” and “Fanplayr Account Key” are required to get Fanplayr to start tracking on your site.

Below is listed details about each option.

Enable Fanplayr? (required)

Set this value to true to enable Fanplayr tracking.

Fanplayr Account Key (required)

You must supply your Account Key to start tracking with Fanplayr. If you do not yet have an Account Key please contact your Fanplayr account manager.

Enable Apply to Cart?

Set to true to have the Fanplayr widget display an “Add to Cart” button, which will try and apply the current coupon and then take the user to the Cart page.

Enable Session Offers?

Fanplayr has the ability to enable coupons only for users who have been given those coupons via Fanplayr, in a visitors session. Please see the section below about Session Offer setup for more details. This defaults to *true*.

Prefix for Session Offers

Session Offers require a prefix to your coupon. Any coupon with this prefix will be disabled for your users unless otherwise enabled by the Fanplayr script. This defaults to "FPTEST_", and should be changed for use in production.

Track during Checkout?

By default this is set to *false*. We do not track during checkout to help you with PCI Compliance. If you would like Fanplayr to track during checkout please enable this option. Please note that we do not send any information about payment methods or details.

Custom JS

Enter JS which will be added to both user and order tracking. Note that double quotes are not supported.

Product Image Viewtype

Override the default of “large” when getting image links for products

Force HTTPS on redirects?

Redirects to Cart (example) are forced from HTTP to HTTPS even when coming from insecure page. Also forces HTTPS for images and product URLs.

Only track 'ORDER_STATUS_NEW' orders?

Only track orders when their status is NEW. Some sites would track failed credit card orders as completed orders.

Add Fanplayr tag to your template

The last step in getting Fanplayr to start tracking is to add our tag to your template. This should be done in a template that is on all pages. Normally this would either be in the header or footer files which should be located:

Header: Templates/components/header/htmlhead.isml

Footer: Templates/components/footer/footer.isml

The tag to include is as follows:

```
<isinclude template="util/FanplayrTracking.isml"/>
```

This template will track for both normal user tracking as well as order tracking.

Test Tracking

At this point it is a good idea to check that Fanplayr is tracking details as we expect. For this you will need to set up a coupon.

Let's set up a coupon as follows for simplicity:

- 10% off cart
- "FPTEST_10OFF" as coupon code

Add an item to your cart, and then make sure the code applies to your cart entered in the normal way ie coupon entry field in the checkout.

You can see what data Fanplayr is tracking by opening the Javascript Console in your favorite browser and inspecting the following variable:

```
window.fanplayr._i[0].data
```

You should then see something like the following:

```
> fanplayr._i[0].data
< ▼ Object {products: Array[1], subTotal: 82, total: 82, discount: 0, discountCode: "..."} ⓘ
  accountKey: "ACCOUNT KEY"
  applyToCartUrl: "Fanplayr-Apply?code=%c"
  categoryId: ""
  categoryName: ""
  currency: "USD"
  discount: 10
  discountCode: "FPTEST_10OFF"
  lineItemCount: 1
  numItems: 1
  pageType: "cart"
  productId: ""
  productImage: ""
  productName: ""
  productPrice: 0
  productSku: ""
  productUrl: ""
  products: Array[1]
    ▼ 0: Object
      id: "701644023942"
      name: "Straight Skirt"
      price: 100
      qty: 1
      sku: "701644023942"
      ▶ __proto__: Object
      length: 1
      ▶ __proto__: Array[0]
      sessionOfferUrl: "Fanplayr-Session?code=%c"
      subTotal: 100
      total: 90
      ▶ __proto__: Object
```

Note that you should check this variable on each of our page types; your home page, category page, product page and the cart. The data that will change is as follows:

- Home page
 - pageType: "home"
- Category page
 - pageType: "cat"
 - categoryId and categoryName set

- Product page
 - pageType: "prod"
 - productId, productImage, productName, productPrice, productSku and productUrl set
- Cart
- pageType: "cart"

You should then complete and order, and on the order confirmation page inspect the Javascript object:

`window.fp_sales_order.data`

You should see something like the following:

```
> fp_sales_order.data
< ▼ Object {products: Array[1], subTotal: 82, total: 82, discount: 0, discountCode: ""...} ⓘ
  accountKey: "ACCOUNT_KEY"
  currency: "USD"
  customerEmail: "info@fanplayr.com"
  customerId: "bfDDTUWlbgzU8a4ldyM7SuLG8c"
  discount: 10
  discountCode: "FPTEST_100FF"
  firstName: "FIRST"
  lastName: "LAST"
  orderEmail: "info@fanplayr.com"
  orderId: "00000501"
  orderNumber: "00000501"
  pageType: "order"
  ▼ products: Array[1]
    ▼ 0: Object
      id: "701644023942"
      name: "Straight Skirt"
      price: 100
      qty: 1
      sku: "701644023942"
      ► __proto__: Object
    length: 1
    ► __proto__: Array[0]
  shipping: 10
  subTotal: 100
  tax: 10
  total: 90
  ► __proto__: Object
```

You should be tracking correctly now.

Test Apply to Cart URL

We will now test to make sure that the Apply to Cart button will work as expected. Do the following:

- Add a product to your cart.
- Go to your site URL suffixed with “/Fanplayr-Apply?code=FPTEST_10OFF”

Note that we are using the same coupon code we used earlier. If you used a different one, please use that code instead.

This should have applied the coupon and taken you to the cart page.

Set up Session Offers

Fanplayr offers the ability to disable coupons, with a specific prefix, from being used until they are given to your users (enabled by Fanplayr). For details of how this is implemented in general please see the following:

<https://fanplayr.gitbooks.io/integration-custom/content/session-offers.html>

In CommerceCloud we have support for Session Offers both using Pipelines and Controllers. Please implement Session Offers using the appropriate method for your site.

Please make sure you change the prefix in preferences (see above) for use in production. By default it is set to "FPTEST_" and only coupons with that prefix will be treated as Session Offers.

Note: the following detailed instructions let you know how to set up Session Coupons based on the GenesisStore. Following the detailed instructions here is information on other places that you may need to implement Session Coupons.

Using Pipeline

Open the Cart Pipeline which is likely at Pipelines/Checkout/Cart. Find the AddCoupon and AddCouponJson sub-pipelines.

You will need to add a Script Pipelet Node between the CouponCode pipeline dictionary variable being set, and it being checked for or applied to cart. You can see where this is added in the following image for the SiteGenesis implementation.

The Script Pipelet needs to run the FanplayrCheckSessionCoupon.ds script.

You will also have to set the two Dictionary Input variables as follows:

- CouponCode > CouponCode
- CurrentSession > CurrentSession

Note that the CouponCode pipeline dictionary variable could be different on your system. If so please replace it with your implemented variable name.

Cart

AddCoupon

Script (bc_api)
Script File: Fanpl...

error

expression Coup...

no

Cart-GetBasket

AddCouponJson

Assign (bc_api)

Script (bc_api)
Script File: Fanpl...

error

Cart-AddCoupon

Properties Problems Console

Pipelet Node - Script (bc_api)

Property	Value
Configuration	
OnError	PIPELET_ERROR
ScriptFile	Fanplayr:FanplayrCheckSessionCoupon.ds
Timeout	
Transactional	false
Dictionary Input	
CouponCode	CouponCode
CurrentSession	CurrentSession
Dictionary Output	
CouponCode	CouponCode
ScriptLog	null
Properties	
Custom Label	
Description	

Using Controller

If you use Controllers for your Cart then please implement Session Offers using the following code. In the SiteGenesis implementation at /controllers/Cart.js please insert the Fanplayr Session Offers Handling Code.

General add coupon

The code should be inserted in the submitForm function between *checking that a coupon value was supplied* and the *calling of the cart.addCoupon method*. Below is where it should be added in the default controller code as of September 2016.

```
if (formgroup.couponCode.htmlValue) {  
    // INSERT FANPLAYR CODE HERE  
    status = cart.addCoupon(formgroup.couponCode.htmlValue);  
}
```

The Session Offers Handling Code is as below:

```
/*  
 * START Fanplayr Session Offers Handling  
 */  
var preferences = dw.system.Site.current.preferences.custom;  
var sessionCouponPrefix : String = preferences.fanplayrSessionCouponPrefix &&  
preferences.fanplayrSessionCouponPrefix.toLowerCase() || null;  
var currentCouponCode = formgroup.couponCode.htmlValue &&  
formgroup.couponCode.htmlValue.toLowerCase();  
  
if ( currentCouponCode && sessionCouponPrefix &&  
currentCouponCode.indexOf(sessionCouponPrefix) === 0 ) {  
    // get allowed session coupons from session  
    var sessionCodesString = session && session.custom &&  
session.custom.fanplayrSessionCodes || '[]';  
    var sessionCodes : Array = new Array();  
    try {  
        sessionCodes = JSON.parse(sessionCodesString);  
    } catch (e) {}  
    // if the session coupon that the user has tried to apply  
    // does not exist in the session then simply remove the code  
    if ( sessionCodes.indexOf(currentCouponCode) === -1 ) {  
        return {  
            cart: cart,  
            CouponError: 'NO_ACTIVE_PROMOTION'  
        };  
    }  
}
```

```
/*  
 * END Fanplayr Session Offers Handling  
 */
```

Add coupon using JSON

Your system may also use Javascript to apply a coupon using JSON, without reloading the page. This is also handled Cart.js controller, inside the addCouponJson function.

The code needs to be added after this line:

```
couponCode = request.httpParameterMap.couponCode.stringValue;
```

And before this call happens:

```
Transaction.wrap(function () {  
    couponStatus = cart.addCoupon(couponCode);  
});
```

The code to insert is almost the same as the above, except instead of *returning* an error, we simply set the couponCode to an empty string if it is not allowed.

```
/*  
 * START Fanplayr Session Offers Handling  
 */  
var preferences = dw.system.Site.current.preferences.custom;  
var sessionCouponPrefix : String = preferences.fanplayrSessionCouponPrefix &&  
preferences.fanplayrSessionCouponPrefix.toLowerCase() || null;  
var currentCouponCode = couponCode && couponCode.toLowerCase();  
  
if ( currentCouponCode && sessionCouponPrefix &&  
currentCouponCode.indexOf(sessionCouponPrefix) === 0 ) {  
    // get allowed session coupons from session  
    var sessionCodesString = session && session.custom &&  
session.custom.fanplayrSessionCodes || '[]';  
    var sessionCodes : Array = new Array();  
    try {  
        sessionCodes = JSON.parse(sessionCodesString);  
    } catch (e) {}
```

```

    // if the session coupon that the user has tried to apply
    // does not exist in the session then simply remove the code
    if ( sessionCodes.indexOf(currentCouponCode) === -1 ) {
        couponCode = '';
    }
}
/*
 * END Fanplayr Session Offers Handling
 */

```

Testing Session Coupons

Having implemented the Session Offer code any code that starts with Session Offer Prefix should now be rejected when a user tried to apply it to the cart. By default the prefix is "FPTEST_" and should be changed in preferences for use in production.

Let's try the following:

- Add a product to your cart.
- Apply the coupon "FPTEST_10OFF" (assuming this is the coupon we used earlier for testing)

This coupon should now fail to be applied.

- Now go to your site URL suffixed with
"/Fanplayr-Session?code=FPTEST_10OFF"
- Apply the coupon "FPTEST_10OFF"

The coupon should now be applied to your cart, given that it is a valid coupon for the user and products.

Session Coupons outside of GenesisStore

The above details how to implement Session Coupons in the GenesisStore, the default implementation that comes with CommerceCloud. It also only implements Session Coupons in the Cart, not during Checkout. The following gives some details on where else you may need to implement Session Coupons.

Using Pipelines

No extra changes should be required here as long as you are calling the Cart-AddCoupon Pipeline. If you are not using this to handle application you will need to add the same FanplayrCheckSessionCoupon script to handle the check for an allowed coupon.

Using Controllers

Some Controllers may need to have code added to them to check for Session Coupons. An example, which may actually be in your GenesisStore, is in the COBilling.js. Use the same code as for adding the coupon with JSON, inserting it in the applyCoupon function between this line:

```
var couponCode = request.httpParameterMap.couponCode.stringValue ||  
request.httpParameterMap.dwfrm_billing_couponCode.stringValue;
```

And before the line:

```
app.getController('Cart').AddCoupon(couponCode);
```