ChannelSight Widget Implementation

Version 1.45

ChannelSight®
Know more. Sell more.

1 Document Info

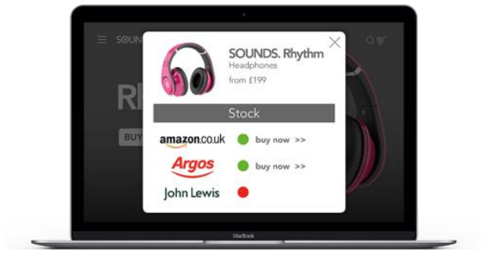
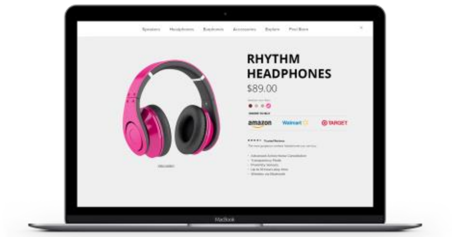
1.1 Document Versions

Version	Author	Date	Notes
1.1 Draft	Moustafa Fawzy	01/02/2019	Initial Document Draft
1.11	Ronan O’Callaghan	11/02/2019	Pending v1.11 release
1.44	Ronan O’Callaghan	2022-01-07	Version bump inline with platform version
1.44	Ben Fairclough	2022-03-31	BannerID retired
1.45	Ben Fairclough	2023-02-22	Updates to tracking parameters

2 Widget Implementation

The implementation consists of adding to a brand page a Widget (**Popup** or **Inline**) that displays the retailers available for a single product or group of products.

This includes two sections of code to be implemented on the brand website depending on the Widget type required:

Pop Up	Inline
 SOUNDS. Rhythm Headphones from £199 Stock amazon.co.uk buy now >> Argos buy now >> John Lewis	
JS Reference: Included in head section of page invoking the asset: <script src="https://cscoreproweustor.blob.core.windows.net/widget/scripts/cswidget.loader.js" defer></script>	**JS Reference:** Included in the head section of page invoking the asset: <script src="https://cscoreproweustor.blob.core.windows.net/widget/scripts/cswidget.loader.js" defer></script>
Widget Container: Include in page accordingly – this part must be implemented on the page where the Popup CTA “Buy Now button” is to be displayed: <button class="cswidget" data-asset-id="XX" data-product-sku="EAN">Buy Now</button>	**Widget Container:** Include in page accordingly - this part must be implemented on the page where the Inline Widget is to be displayed: <div class="cswidget" data-asset-id="**XX**" data-product-sku=**"EAN"**></div>

The attributes that need to be amended within the code are the following:

data-asset-id: is the widget identifier for ChannelSight and it has a static value. The asset ID determines the list of retailers, ordering, configuration and any styling applied within each widget.

NOTE: *For deployments utilising multiple widgets within the same site, across multiple sites or implementing a single widget across multiple sites, please see the advanced section within the Appendix below for more info on widget setup and tracking across widgets.*

data-product-sku: is the product ID corresponding to a specific product or a list of product IDs where the widget will display multiple products. In the latter case the product IDs must be separated by coma (**,**).

ChannelSight recommends the values of the **data-product-sku** attribute to be passed dynamically to the widget via a site's content management system in order to ensure that on each product page the corresponding widget will display the associated products automatically.

The only scenario where the **data-product-sku** should be static (fixed) is where it will not change, on a fixed landing or campaign page for example. This can be configured directly within the widget or within the above code.

NOTE: *If a product that is not within the product catalogue is referenced within a widget, the widget will not load any product(s).*

Please find an example below displaying a basic HTML file invoking a Popup Widget with multiple products:

```
<html>
<head>
<title>Title</title>
<script src="https://cscoreproweust                    dget/scripts/cswidget.loader.js" defer></script>
</head>
<body>
<button class="cswidget" data-asset-id="1234" data-product-sku="DemoSKU1,DemoSKU2,DemoSKU3">Buy
Now</button>
</body>
</html>
```

Unique ID per Widget

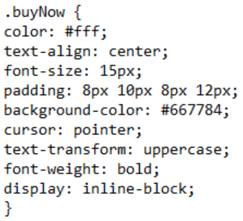
Product ID - Dynamic

Appendix: Advanced Implementation Options

Buy Button:

The Buy Button that will appear on the brand product page has a very simple styling applied to it that can be further customized or changed based on your preferences.

Default Buy Button Styling:



```
.buyNow {
color: #fff;
text-align: center;
font-size: 15px;
padding: 8px 10px 8px 12px;
background-color: #667784;
cursor: pointer;
text-transform: uppercase;
font-weight: bold;
display: inline-block;
}
```

Implementing multiple Widgets (multiple stylings or brands) within a single Site:

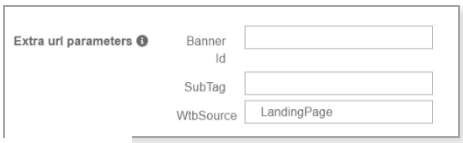
Where you want to display multiple widgets on the same site to display different product assortments or different brands then you will need to configure a Widget for each brand or product assortment depending on the retailer configuration and styling of each widget. Each widget will have a unique asset ID based on the configuration applied per widget.

***Note**: Each Widget configured for a specific brand must only be configured with products for that brand.*

Implementing a single Widget across multiple Sites:

If you would like to publish the same widget across more than one site then you have two options:

1) Creating two similar widgets within the portal and then configuring an extra tracking parameter to determine where each widget will be published (Figure 1.3).



Extra url parameters 🛈

Banner Id	
SubTag	
WtbSource	LandingPage

Figure 1.3

This can easily be done by configuring the **"WtbSource"** parameter within the "Feed Options" section in the portal (figure 1.3) which then determines the location of each widget.

2) The second and recommended option is to create a single widget, publish it on any number of sites and then either statically or dynamically pass a tracking parameter into each widget, which in turn reports on the location for that widget.

Please note: Any manual or dynamic addition of tracking parameters will override the parameters configured within the portal (figure 1.3). This can be done as per the below overview **"Widget Tracking"**.

Widget Tracking:

Appending additional attributes to an existing widget

ChannelSight offers the ability to track and report on traffic and conversions generated by different campaigns distinctly. It is also capable of joining data sets within different reporting suites in addition to reporting on different platforms where the service is active.

This is achieved by appending individual attributes to the ChannelSight Widget Container. See below for the list of available attributes and there intended usage.

Attribute	Use for	Reporting
data-csi_utm_source	channel	Used to aggregate channel performance and filter campaigns
data-csi_utm_medium	website	Used to aggregate websites and filter campaigns
data-csi_utm_campaign	Campaign name	Used as a reporting dimension to isolate individual campaigns
data-csi_utm_content	Ad-type	Used as a reporting dimension to isolate individual ad-types
data-subTag	Click/sessionID/data-key	Not used for reporting, but can be exported
data-wtbsource	Page-types	Used as a reporting dimension to differentiate page-types

Please note:

- *Any manual or dynamic addition of tracking parameters will override the parameters configured within the portal (figure 1.3).*
- *Where a parameter contains spaces or special character such as **"&"** or **"="** those values need to be URL encoded in order to properly record the values in the CS system.*

Appending Campaign Tags.

Campaign Tracking

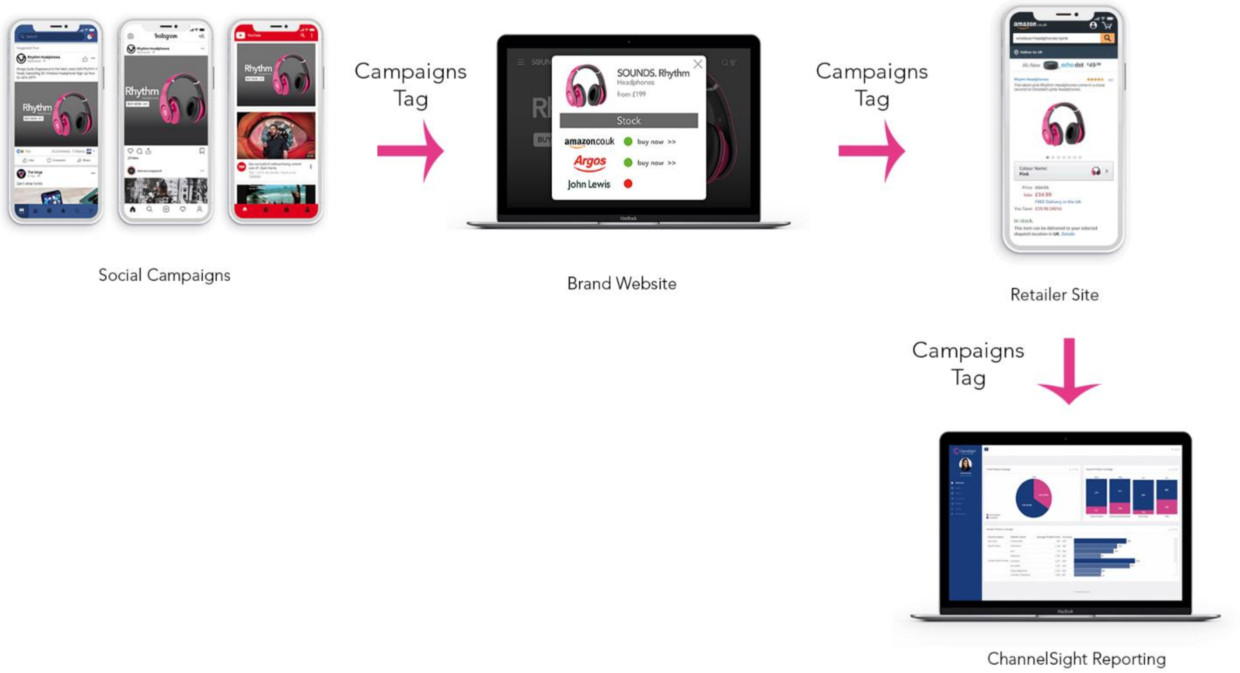
Where you are running different campaigns generating traffic to your website and you wish to measure the impact of each campaign on the Buy Now service, then you can do this by dynamically appending the campaign info to the widgets upon each page load.

This will then allow you to differentiate between traffic from campaign A vs campaign B and determine the effectiveness of each campaign by comparing the clicks and product sales related to each campaign via our reporting suite.

This is done by utilising the data-csi_utm_***fields***. Dynamically identifying and passing any campaign terms to the widget container, which will then in turn be passed through with any click to the retailer site, associated with any sales, and reported on within the ChannelSight reporting suite.

Example:

```
<button class="cswidget" data-asset-id="1234" data-product-sku="DemoSKU1,DemoSKU2,DemoSKU3" data-csi_utm _medium=social data-csi_utm_source=facebook data-csi_utm_campaign=spring20 data-csi_utm_content=video>Buy Now</button>
```



Appending Unique-tracking Codes

Where you have a reporting suite that tracks users throughout your website(s) by assigning anonymous session or user ID’s to each user/session then that same session or user ID can be fetched by ChannelSight, by adding the container using the **data-cookie-consent** attribute.

Example:

```
<button class="cswidget" data-asset-id="1234" data-product-sku="DemoSKU1,DemoSKU2,DemoSKU3" data-cookie-consent=true>Buy Now</button>
```

Notes on dynamic subtag:

- UserIDs that are supported are specific to Adobe & Google Analytics ; AMCV, _ga & _gid
- Cookie consent is determined by the brand, as a data-processor, ChannelSight is authorized to process data on behalf of the data-processor (the client). GDPR compliance with measuring personal-data can be determined in two ways using this feature :
    - Hard-set parameter to ‘**cookie-consent=true’** – ChannelSight will fetch an accepted user-ID where available regardless of the users cookie-consent. Client then needs to filter user-IDs that did or did-not provide consent when the data is imported.
    - Client defines user cookie consent **‘cookie-consent=true/false’** here the widget call includes whether the user has accepted or rejected your cookie-consent policy and provides permission to ChannelSight to fetch the user-ID, or to do nothing.

Appending Platform Identifiers

Landing Page

After creating the widget for the website and publishing it on the product page(s) you determine that you’d like to also use the same widget on a stand-alone landing page. As this will be for a static landing page and the widget won’t necessarily need to be configured in the future you can append the “csi_wtbsource” parameter within the source code to then differentiate within our reporting platform the clicks and sales that originated from product pages vs the static landing page.

Product Page Widget Code	<button class="cswidget" data-asset-id="01234" data-product-sku="DemoSKU1,DemoSKU2,DemoSKU3">Buy Now</button>
Landing Page Widget Code	<button class="cswidget" data-asset-id="01234" data-product-sku="DemoSKU1,DemoSKU2,DemoSKU3" ***data-wtbSource=LandingPage***>Buy Now</button>