

# TwicPics SFRA Integration Guide

Version 21.1.0



# Table of Contents

<b>1. Version .....</b>	<b>3</b>
<b>2. Terms and Acronyms .....</b>	<b>3</b>
<b>3. About TwicPics .....</b>	<b>4</b>
<b>4. About this guide .....</b>	<b>4</b>
<i>Supported features .....</i>	<i>4</i>
<b>5. Component Overview .....</b>	<b>4</b>
<i>Functional Overview .....</i>	<i>4</i>
<i>Limitations, Constraints .....</i>	<i>4</i>
<i>Compatibility .....</i>	<i>5</i>
<i>Privacy, Payment .....</i>	<i>5</i>
<i>Twicpics cartridge .....</i>	<i>5</i>
<b>6. Implementation Guide .....</b>	<b>5</b>
<i>Installation .....</i>	<i>5</i>
Build Client CSS files .....	5
Set Cartridge Path .....	6
Import System Object .....	6
Site Preferences configuration .....	7
TwicPics configuration .....	9
<i>Custom Code .....</i>	<i>12</i>
<b>7. Testing .....</b>	<b>14</b>
<b>8. Code changes .....</b>	<b>14</b>
<i>int_twicpics .....</i>	<i>14</i>
<i>int_twicpics_sfra .....</i>	<i>15</i>
<i>twicpics_sfra_changes .....</i>	<i>15</i>
<b>9. Operations, Maintenance .....</b>	<b>28</b>
<i>Availability .....</i>	<i>28</i>
<i>Failover/Recovery Process .....</i>	<i>28</i>
<i>Support .....</i>	<i>28</i>
<b>10. Release history .....</b>	<b>29</b>

## 1. Version

Version Type	Version
<b>Current Stable Version</b>	v21.1.0
<b>SFRA Version</b>	v6.0.0

## 2. Terms and Acronyms

Acronym	Description
<b>Salesforce B2C Commerce (Cloud)</b>	Salesforce Business to Client Commerce (Cloud)
<b>SFCC</b>	Salesforce Commerce Cloud
<b>BM</b>	Business Manager – Admin panel of Salesforce B2C Commerce
<b>Storefront</b>	The e-commerce solution of a website hosted on Salesforce B2C Commerce platform.
<b>SFRA</b>	Storefront Reference Architecture – new Salesforce B2C Commerce Storefront Architecture
<b>int_twicpics</b>	The main TwicPics Salesforce Commerce Cloud cartridge which will be used as basis for any Merchant Storefront version
<b>int_twicpics_sfra</b>	Twicpics Salesforce Commerce Cloud extension cartridge for SFRA based storefront. All changes in this cartridge are made based on Storefront Reference Architecture.

## 3. About TwicPics

TwicPics is a real-time image processing service that enables businesses of all sizes to deliver high performing, rich visual content with an easy setup. It is a responsive image solution (SaaS). It offers on-demand responsive image generation combined with a smart, unobtrusive JavaScript library, all powered by a no-nonsense, testable URL based API.

Developers only deal with high-resolution versions of their media while end-users receive optimized, perfectly sized, device-adapted versions delivered from a server close to them.

Learn more about TwicPics on [our website](#).

## 4. About this guide

This document provides technical instructions to integrate into an SFRA code base the TwicPics Salesforce Commerce Cloud cartridge.

### Supported features

- Override product images and swatch colors with TwicPics images
- Convert all content, static and site images into TwicPics images
- Automatic layout-driven sizing & cropping
- Lazy Loading out of the box
- SEO Optimizations

## 5. Component Overview

### Functional Overview

This cartridge enables TwicPics images to be displayed on an SFRA-based storefront.

### Limitations, Constraints

Using the TwicPics cartridge requires credentials, URLs and mapping from TwicPics. Please go to the TwicPics website to [sign up for the services](#).

## Compatibility

Designed for SFRA version 6.0.0. Compatibility Mode 21.2. TwicPics cartridge is working with all locales.

## Privacy, Payment

No user or billing information are collected by using the cartridge.

## Twicpics cartridge

- **Core business logic cartridge:** The main cartridge that contains all core business logic of TwicPics is **int\_twicpics**. This cartridge should be added to any of your sites supposed to use TwicPics functionalities.
- **Extension cartridge:**
  - **int\_twicpics\_sfra** cartridge should be used with **int\_twicpics** cartridge for SFRA based storefront. This cartridge extends any model/controller/client file that needs to be adapted to TwicPics logic.
  - **twicpics\_sfra\_changes** cartridge is used with **int\_twicpics\_sfra** and **int\_twicpics** cartridges. It contains all SFRA templates that were adapted for the use of TwicPics.

## 6. Implementation Guide

### Installation

The installation of the TwicPics Salesforce Commerce Cloud cartridge is easy and requires just a few steps: upload the TwicPics cartridges, add them to the cartridge path of your site, import metadata, set configurations in your site preferences and configure your TwicPics account.

### Build Client JS and CSS files

Before being able to run any node npm command in your terminal, you have to install node.js (the latest stable version is recommended). Once the installation is completed, please run the following command in the **twicpics** folder:

```
npm install; npm run compile:scss; npm run compile:js
```

## Set Cartridge Path

- In the Business Manager, go to **Administration > Sites > Manage Sites**
- Select your site
- Click on the **Settings** tab and add **twicpics\_sfra\_changes**, **int\_twicpics** and **int\_twicpics\_sfra** cartridges on the left of the **app\_storefront\_base** cartridge.

Your cartridge path should look like that:

Cartridges:	twicpics_sfra_changes:int_twicpics_sfra:int_twicpics:app_storefront_base
Effective Cartridge Path:	twicpics_sfra_changes int_twicpics_sfra int_twicpics app_storefront_base

- Click **Apply**

## Import System Object

- In the Business Manager, go to **Administration > Site Development > Import & Export**
- Click **Upload**
- Click **Choose a file**
- Select the file **meta/system-objecttype-extensions.xml**
- Click **Upload**

[Administration](#) > [Site Development](#) > [Import & Export](#) > Manage Import Files

## Upload Import Files

Upload File:	<input type="button" value="Choisir un fichier"/> system-object...extensions.xml <input type="button" value="Upload"/>
--------------	--

- Go to **Administration > Site Development > Import & Export**
- Click **Import** in the **Meta Data** part

### Meta Data

[Import](#) and [export](#) your system meta data (i.e., system type extensions, custom object types, custom preference definitions).

- Select **system-objecttype-extensions.xml**
- Click **Next >>**
- Once the file validation is done, click **Import**

## Site Preferences configuration

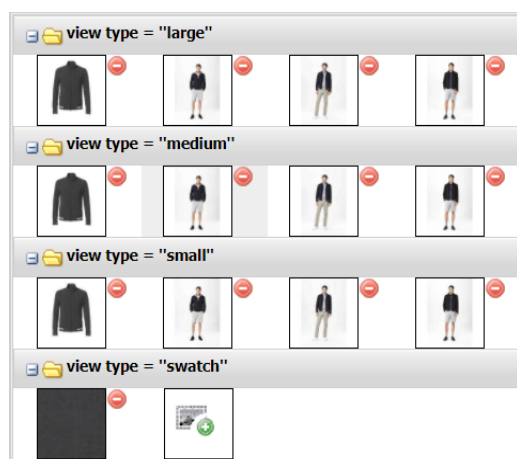
- In the BM, go to **Merchant Tools > Site Preferences > Custom Preferences**
- Click on **Twicpics**
- Fill the different fields as follow

ID	Description	Value (example)
<b>TWICEnabled</b>	Enable or disable the TwicPics cartridge and select the TwicPics mode	<ul style="list-style-type: none"><li>• <b>None:</b> the cartridge is disabled</li><li>• <b>Standard:</b> TwicPics images are displayed on the storefront</li><li>• <b>Advanced:</b> TwicPics images are displayed on the storefront and a TwicPics JS script is enabled</li><li>• <b>Expert:</b> TwicPics images are displayed on the storefront, a TwicPics JS script is enabled and &lt;img&gt; tags are processed via a TwicPics module to optimize the render. Expert mode is compliant with Page Designer.</li></ul>
<b>TWICHostname</b>	Your own Twicpics URL. Find your id by connecting to your <a href="#">TwicPics account</a> . In the Domain tab.	https://[YourTwicPicsID].twic.pics
<b>TWICScriptVersion</b>	Refer to <a href="#">Script: Set Up   TwicPics Documentation</a>	v1

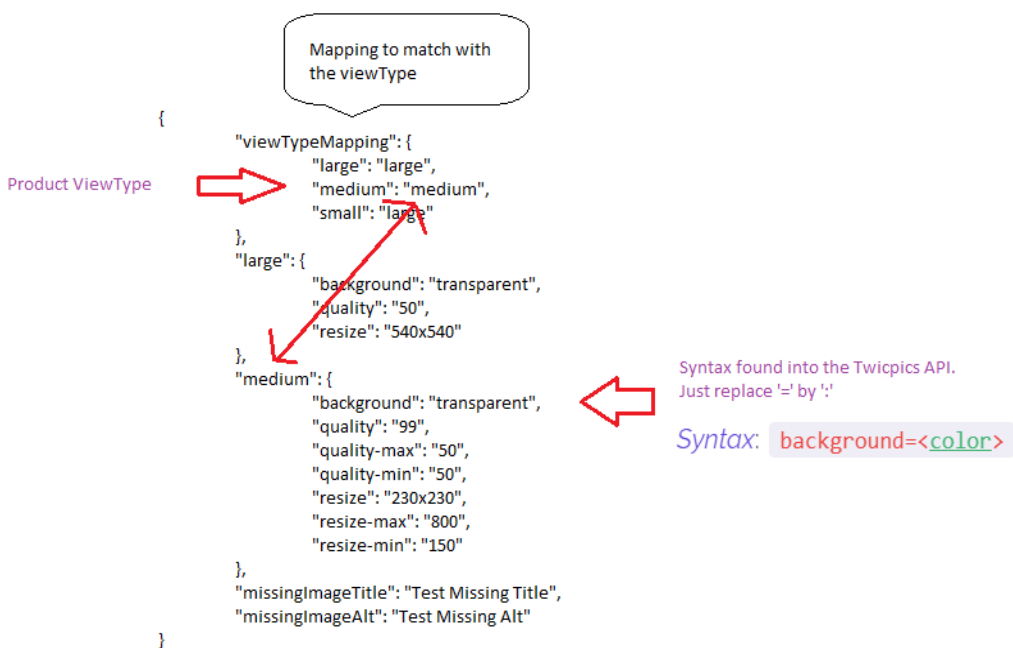
ID	Description	Value (example)
<b>TWICStaticPath</b>	Path for static images mapping on Twicpics. It must be consistent with the path defined on your TwicPics account (later in this document).	static
<b>TWICContentPath</b>	Path for content images mapping on Twicpics. It must be consistent with the path defined on your TwicPics account (later in this document).	content
<b>TWICCatalogPath</b>	Path for catalog images mapping on Twicpics. It must be consistent with the path defined on your TwicPics account (later in this document). This field is a set of string and must contain all master catalog IDs.	electronics-m-catalog apparel-m-catalog
<b>TWICSitePath</b>	Path for site images mapping on Twicpics. It must be consistent with the path defined on your TwicPics account (later in this document).	site

ID	Description	Value (example)
<b>TWICTransaformations</b>	<p>JSON configuration to render TwicPics images.</p> <p>See below for more information about the JSON construction.</p> <p>To see all values available, go on: <a href="#">API: Transformations   TwicPics Documentation</a></p> <p>As you can see in the opposite example, for the <b>viewTypeMapping</b> property, you can map "small" to "large" or anything else as your convenience.</p>	<pre>{   "viewTypeMapping": {     "large": "large",     "medium": "medium",     "small": "large"   },   "large": {     "background": "transparent",     "quality": "50",     "resize": "540x540"   },   "medium": {     "background": "transparent",     "quality": "99",     "quality-max": "50",     "quality-min": "50",     "resize": "230x230",     "resize-max": "800",     "resize-min": "150"   },   "missingImageTitle": "Test Missing Title",   "missingImageAlt": "Test Missing Alt" }</pre>

You can use view types for all kind of images: CATALOG, CONTENT, STATIC or SITE



Different view types available in SFCC



JSON structure for TWICTransformations field

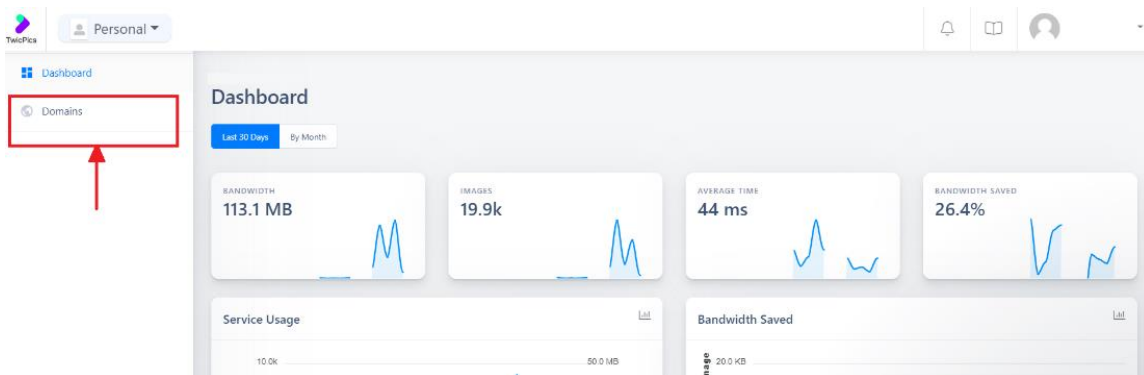


## TwicPics configuration

- Create an account on the [Twicpics site](#)
- Confirm your account by clicking the link in the email you received

Now we have to create new mapping paths that are consistent with values **TWICStaticPath**, **TWICCatalogPath**, **TWICContentPath** and **TWICSitePath** filled earlier in the BM (in the TwicPics custom site preference). *To find these values in the BM, go to **Merchant Tools > Site Preferences > Custom Preferences > Twicpics**.*

- Go back to your [TwicPics account](#)
- Click the **Domains** tab



- Click **Add new path** button
- Fill the **Path** field with the same value as **TWICStaticPath**. Pay attention the value must be identical and is case sensitive.
- Fill the **Source URL** field with this value: [https://\[host\]/on/demandware.static/Sites-\[site\\_id\]-Site/-/](https://[host]/on/demandware.static/Sites-[site_id]-Site/-/), replace **[host]** by your SFCC hostname and **[site-id]** by your site ID.

*If for example in the BM you set **TWICStaticPath** value to **static**, the form should be filled like this :*

The image shows a configuration form for TwicPics. It has two main input fields: 'Path' and 'Source URL'. The 'Path' field contains the text 'https://[redacted].twicpics/static'. Below it, a note says 'Path will point to the remote directory specified in Source URL.' with a yellow checkmark icon. The 'Source URL' field contains the text 'https://[redacted]/on/demandware.static/Sites-RefArch-Site/-/'. Below it, a note says 'Source URL specifies the remote directory Path will point to.' with a yellow checkmark icon. A red arrow points to the 'RefArch' part of the Source URL, with the text 'Replace RefArch by your Site ID' written above it.

- Click **Apply**

Same operation for content images:

- Click **Add new path** button
- Fill the **Path** field with the same value as **TWICContentPath**. Pay attention the value must be identical and is case sensitive.
- Fill the **Source URL** field with this value: [https://\[host\]/on/demandware.static/-/Library-Sites-\[library-id\]/](https://[host]/on/demandware.static/-/Library-Sites-[library-id]/), replace **[host]** by your SFCC hostname and **[library-id]** by the library ID.

If for example in the BM you set **TWICContentPath** value to **content**, the form should be filled like this :

Path

https://[redacted]twic.pics/content/ /

Path will point to the remote directory specified in Source URL. ☺

Source URL

Replace RefArchSharedLibrary by your library ID

https://[redacted]/on/demandware.static/-/Library-Sites-RefArchSharedLibrary/

Source URL specifies the remote directory Path will point to. ☺

- Click **Apply**
- Repeat these 4 last steps for each library you want to add

Same operation for site images:

- Click **Add new path** button
- Fill the **Path** field with the same value as **TWICSitePath**. Pay attention the value must be identical and is case sensitive.
- Fill the **Source URL** field with this value: [https://\[host\]/on/demandware.static/-/Sites/](https://[host]/on/demandware.static/-/Sites/), replace **[host]** by your SFCC hostname.

If for example in the BM you set **TWICSitePath** value to **site**, the form should be filled like this :

Path

https://[redacted]twic.pics/site/ /

Path will point to the remote directory specified in Source URL. ☺

Source URL

https://[redacted]/on/demandware.static/-/Sites/

Source URL specifies the remote directory Path will point to. ☺

- Click **Apply**

Same operation for catalog images:

- Click **Add new path** button
- Fill the **Path** field with the same value as one of those contained in **TWICCatalogPath**. Pay attention the value must be identical and is case sensitive.
- Fill the **Source URL** field with this value: [https://\[host\]/on/demandware.static/-/Sites-\[catalog\\_id\]/default/](https://[host]/on/demandware.static/-/Sites-[catalog_id]/default/), replace **[host]** by your SFCC hostname and **[catalog\_id]** by the catalog id.

If for example in the BM you set **TWICCatalogPath** value to **apparel-m-catalog**, the form should be filled like this :

Path

https://[redacted]twic.pics/ apparel-m-catalog /

Path will point to the remote directory specified in Source URL. ☺

Source URL

Replace apparel-m-catalog by your catalog ID

https://[redacted]on/demandware.static/-/Sites-apparel-m-catalog/default/

Source URL specifies the remote directory Path will point to. ☺

- Click **Apply**
- Repeat these four last steps for each catalog path you defined in **TWICCatalogPath**

## Custom Code

In order to display Twicpics images on different pages, you need to include custom code in your templates. For reference site you may use the **twicpics\_sfra\_changes** cartridge, custom code is already available in it. Don't hesitate to refer to the **twicpics\_sfra\_changes** code as an example to integrate the TwicPics solution to your storefront.

Twicpics images are rendered via the **Twicpics.js** controller contained in the **int\_twicpics** cartridge. So, in your templates, you have to replace **<img>** tags by **<isinclude>** tags to call the **Twicpics** controller on images where you want to use TwicPics for the render. The **Twicpics** controller will then render an **<img>** tag in the html with the TwicPics image.

For example, if you have an image declared as follow in your isml:

```
<img
  class="img"
  src=="image.png"
  alt="My super image"
  title="My super image"
>
```

It will become an isinclude tag:

```
<isinclude url="${URLUtils.url(
  'Twicpics-Image',
  'class', 'img',
  'src', 'image.png',
  'alt', 'My super image',
  'title', 'My super image',
  'imageType', 'CATALOG',
  'width', '32',
  'height', '130',
)}"/>
```

The remote include calls the **Twicpics-Image** controller with all the **img** tag attributes as parameters. In the example above, we can retrieve in the **isinclude** tag, the **class**, **src**, **alt** and **title** attributes from the **img** tag.

The **isinclude** tag also contains **imageType**, **width** and **height** attributes. **imageType** indicates the type of the image (it could be **CATALOG**, **SITE**, **CONTENT** or **STATIC**), it is required to generate a proper TwicPics url. The size attributes **width** and **height** are useful only when an image is not squared to get a correct image ratio.

Here is the list of the attributes supported by Twicpics when creating a TwicPics **<img>** tag via a remote include (all other attributes will be ignored):

Attribute	Role	Specifications
<b>src</b>	Image source url	<ul style="list-style-type: none"> <li>Required</li> </ul>
<b>imageType</b>	Type of the image (CATALOG, STATIC, SITE or CONTENT)	<ul style="list-style-type: none"> <li>Required</li> </ul>
<b>class</b>	<img> tag class attribute	<ul style="list-style-type: none"> <li>Not required</li> </ul>
<b>alt</b>	<img> tag alt attribute	<ul style="list-style-type: none"> <li>Not required</li> </ul>
<b>style</b>	Style attribute is used to apply style in the <img> tag	<ul style="list-style-type: none"> <li>Not required</li> </ul>
<b>height</b>	Height of the image	<ul style="list-style-type: none"> <li>Not required in Standard and Advanced mode</li> <li>Required on not squared images in expert mode</li> </ul>
<b>width</b>	Width of the image	<ul style="list-style-type: none"> <li>Not required in Standard and Advanced mode</li> <li>Required on not squared images in expert mode</li> </ul>
<b>title</b>	Title of the image	<ul style="list-style-type: none"> <li>Not required</li> </ul>
<b>data-index</b>	Index of the image displayed – used in reference site	<ul style="list-style-type: none"> <li>Not required</li> </ul>
<b>item-prop</b>	Item property (image for ex.) – used in reference site	<ul style="list-style-type: none"> <li>Not required</li> </ul>
<b>placeholder</b>	Type of display of the background image. Values are 'none', 'preview', 'meancolor' or 'maincolor'. Default is 'preview'	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>
<b>ratio</b>	Used to compute the padding ratio of the image	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>
<b>step</b>	To avoid requesting too many variants of the same medium (as could happen in case of 'width: 100%' elements for instance), the Script will round the width of media to the closest multiple of 'step'. The height will then be computed so as to respect the original aspect ratio. You can change this value to get as close or as far from a pixel-perfect resizing as desired. Default value is '10'. Any value that is not a number will be ignored.	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>
<b>focus</b>	Specifies the focus point for transformations. The value of this attribute must be 'none' or valid coordinates or 'auto'. Default value is 'none'	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>
<b>transition</b>	Whether or not to load images with a fade in effect. 'transition' is a Boolean and default value is true	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>
<b>transitionDelay</b>	Transition delay of the transition effect. Default is '0ms'	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>
<b>transitionDuration</b>	Duration of the effect. Default is '400ms'	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>
<b>transitionTimingFunction</b>	CSS timing function applied to the transition effect. Default is 'ease'	<ul style="list-style-type: none"> <li>Not required</li> <li>Only available in Expert mode</li> </ul>

## 7. Testing

Once the cartridge is installed and templates overridden, just go on the storefront to check if images are generated by TwicPics by inspecting the element. In **Standard**, **Advanced** and **Expert** mode, all images are generated via Twicpics and page designer is taken into account. If it's missing a contribution, a custom TwicPics log file is available in the Business Manager.

## 8. Code changes

### int\_twicpics

Here is a quick explanation of the role of each file included in the **int\_twicpics** cartridge.

#### Controller Twicpics.js

The controller **Twicpics-Image** creates an image object via a model. This object contains different attributes depending on the activation of TwicPics or not and on the TwicPics mode selected. This TwicPics image object is then passed to the view and the image is rendered by **twicpics.isml** template.

#### Model twicimg.js

The model **twicimg** creates a twicpics image object when in **Expert** mode. It enables attributes that will later be interpreted by the TwicPics script.

#### Template twicpics.isml

This template renders an **<img>** tag to the html. Attributes are displayed depending on the TwicPics mode and the image object passed to the view.

#### Helpers

Helpers **ImageTwicpics.js** and **twicpicsHelpers.js** contain functions to help the construction of the TwicPics image (get TwicPics url, get image transformations...).

#### Template twicpicsHeader.isml

This template is used to run the TwicPics script.

## Hook **htmlhead.js**

This hook is used to render **twicpicsHeader.isml** template when in **Advanced** and **Expert** mode.

## int\_twicpics\_sfra

Here are all the **SFRA** files that have been extended to enable TwicPics.

## Model **productImages.js**

Model **productImages** was overridden to get TwicPics images instead of basic images.

## twicpics\_sfra\_changes

This cartridge contains the SFRA files that have been overridden for the good use of TwicPics on reference sites

## Template **htmlHead.isml**

This template was overridden to add a link to load TwicPics CSS.

Line	int_twicpics_sfra
33	<pre>&lt;link rel="stylesheet" href="{URLUtils.staticURL('/css/twicpics.css')}" /&gt;</pre>

## Templates with images

All templates with **<img>** tags were overridden to enable TwicPics images. **<img>** tags were replaced by **<isinclude>** tags to call the **Twicpics-Image** controller used to generate TwicPics **<img>** tags.

Here are the exact changes that occurred in 24 templates.

### orderHistoryCard.isml

Line	storefront_reference_architecture cartridge
21	<pre>&lt;img     class="img-fluid"     src="\${order.firstLineItem.imageUrl}"     alt="\${order.firstLineItem.alt}"     title="\${order.firstLineItem.title}" &gt;</pre>

Line	int_twicpics_sfra cartridge
21	<pre>&lt;isinclude url="\${URLUtils.url(</pre>
22	<pre>    'Twicpics-Image',</pre>
23	<pre>    'class', 'img-fluid',</pre>
24	<pre>    'src', order.firstLineItem.imageUrl,</pre>
25	<pre>    'alt', order.firstLineItem.alt,</pre>
26	<pre>    'title', order.firstLineItem.title,</pre>
27	<pre>    'imageType', 'CATALOG'</pre>
28	<pre>))"/&gt;</pre>

### campaignBanner.isml

Line	storefront_reference_architecture cartridge
10	<pre>&lt;img src="\${URLUtils.staticURL('/images/close-icon.svg')}" /&gt;</pre>

Line	int_twicpics_sfra cartridge
10	<pre>&lt;isinclude url="\${URLUtils.url(</pre>
11	<pre>    'Twicpics-Image',</pre>
12	<pre>    'src', URLUtils.staticURL('/images/close-icon.svg'),</pre>
13	<pre>    'imageType', 'STATIC'</pre>
14	<pre>))"/&gt;</pre>



## cartNestedBonusProductCard.isml

Line	storefront_reference_architecture cartridge
10	<code>&lt;img class="product-image"</code>
11	<code>src="\${productLineItem.images.small[0].url}"</code>
12	<code>alt="\${productLineItem.images.small[0].alt}"</code>
13	<code>title="\${productLineItem.images.small[0].title}"</code>
14	<code>&gt;</code>

Line	int_twicpics_sfra cartridge
10	<code>&lt;isinclude url="\${URLUtils.url(</code>
11	<code>    'Twicpics-Image',</code>
12	<code>    'class', 'product-image',</code>
13	<code>    'src', productLineItem.images.small[0].url,</code>
14	<code>    'alt', productLineItem.images.small[0].alt,</code>
15	<code>    'title', productLineItem.images.small[0].title,</code>
16	<code>    'imageType', 'CATALOG'</code>
17	<code>)}"/&gt;</code>

## imageCarousel.isml

Line	storefront_reference_architecture cartridge
15	<code>&lt;img</code>
	<code>src="\${image.url}"</code>
	<code>class="d-block img-fluid"</code>
	<code>alt="\${image.alt} image number \${image.index}"</code>
	<code>itemprop="image"</code>
	<code>/&gt;</code>

Line	int_twicpics_sfra cartridge
15	<code>&lt;isif condition="\${pdict.isShowQuickView}"&gt;</code>
16	<code>    &lt;isset name="twicpicsImg" value="\${image.twicpicsImg}" scope="page" /&gt;</code>
17	<code>    &lt;isinclude template="twicpics"/&gt;</code>
18	<code>&lt;iselse/&gt;</code>
19	<code>    &lt;isinclude url="\${URLUtils.url(</code>
20	<code>        'Twicpics-Image',</code>
21	<code>        'class', 'd-block img-fluid',</code>
22	<code>        'src', image.url,</code>
23	<code>        'alt', image.alt + ' image number ' + image.index,</code>
24	<code>        'itemprop', 'image',</code>
25	<code>        'imageType', 'CATALOG'</code>
26	<code>    )}"/&gt;</code>
27	<code>&lt;/isif&gt;</code>

## photoTile.isml

Line	storefront_reference_architecture cartridge
12	<code>&lt;img</code>
13	<code>  class="photo-tile-image image-fluid"</code>
14	<code>  src="\${pdict.image.src.mobile}"</code>
15	<code>  style="--focal-point-x:\${pdict.image.focalPointX}; --focal-point-</code>
16	<code>y:\${pdict.image.focalPointY}"</code>
16	<code>  &lt;isif condition="\${pdict.image.alt}"&gt;alt="\${pdict.image.alt}" title="\${pdict.image.alt</code>
17	<code>}"&lt;/isif&gt;</code>
17	<code>/&gt;</code>

Line	int_twicpics_sfra cartridge
12	<code>&lt;isinclude url="\${URLUtils.url(</code>
13	<code>  'Twicpics-Image',</code>
14	<code>  'class', 'photo-tile-image image-fluid',</code>
15	<code>  'src', pdict.image.src.mobile,</code>
16	<code>  'alt', pdict.image.alt ? pdict.image.alt : '',</code>
17	<code>  'title', pdict.image.alt ? pdict.image.alt : '',</code>
18	<code>  'style', '--focal-point-x:'+pdict.image.focalPointX+'; --focal-point-</code>
19	<code>y:'+pdict.image.focalPointY,</code>
20	<code>  'imageType', 'CONTENT',</code>
21	<code>  'placeholder', 'none'</code>
21	<code>  'width', '534',</code>
21	<code>  'height', '233'</code>
21	<code>)}"/&gt;</code>

## StoredPaymentInstruments.isml

Line	storefront_reference_architecture cartridge
4	<code>&lt;img class="card-image \${loopState.first ? 'checkout-hidden' : ''}"</code>
5	<code>  src="\${paymentInstrument.cardTypeImage.src}"</code>
6	<code>  alt="\${paymentInstrument.cardTypeImage.alt}"</code>
7	<code>&gt;</code>

Line	int_twicpics_sfra cartridge
4	<code>&lt;isinclude url="\${URLUtils.url(</code>
5	<code>  'Twicpics-Image',</code>
6	<code>  'class', loopState.first ? 'card-image checkout-hidden' : 'card-image',</code>
7	<code>  'src', paymentInstrument.cardTypeImage.src,</code>
8	<code>  'alt', paymentInstrument.cardTypeImage.alt,</code>
9	<code>  'placeholder', 'none',</code>
10	<code>  'imageType', 'STATIC'</code>
11	<code>)}"/&gt;</code>

## uncategorizedProductCard.isml

Line	storefront_reference_architecture cartridge
18	<pre>&lt;img   class="product-image"   src="\${lineItem.images.small[0].url}"   alt="\${lineItem.images.small[0].alt}"   title="\${lineItem.images.small[0].title}" &gt;</pre>

Line	int_twicpics_sfra cartridge
18	<pre>&lt;isinclude url="\${URLUtils.url(</pre>
19	<pre>    'Twicpics-Image',</pre>
20	<pre>    'class', 'product-image',</pre>
21	<pre>    'src', lineItem.images.small[0].url,</pre>
22	<pre>    'alt', lineItem.images.small[0].alt,</pre>
23	<pre>    'title', lineItem.images.small[0].title,</pre>
24	<pre>    'imageType', 'CATALOG'</pre>
25	<pre>))"/&gt;</pre>

## productListTile.isml

Line	storefront_reference_architecture cartridge
15	<pre>&lt;img</pre>
16	<pre>  class="ITC-image image-fluid common-image-component"</pre>
17	<pre>  src="\${pdict.image.src.mobile}"</pre>
18	<pre>  style="--focal-point-x:\${pdict.image.focalPointX}; --focal-point-</pre>
19	<pre>y:\${pdict.image.focalPointY}; max-height:250px"</pre>
19	<pre>  alt="\${pdict.alt ? pdict.alt : imageName}" title="\${pdict.alt ? pdict.alt : imageName}"</pre>
20	<pre>/&gt;</pre>

Line	int_twicpics_sfra cartridge
15	<pre>&lt;isinclude url="\${URLUtils.url(</pre>
16	<pre>    'Twicpics-Image',</pre>
17	<pre>    'class', 'ITC-image image-fluid common-image-component',</pre>
18	<pre>    'src', pdict.image.src.mobile,</pre>
19	<pre>    'alt', pdict.alt ? pdict.alt : imageName,</pre>
20	<pre>    'title', pdict.alt ? pdict.alt : imageName,</pre>
21	<pre>    'style', '--focal-point-x:'+pdict.image.focalPointX+'; --focal-point-</pre>
22	<pre>y:'+pdict.image.focalPointY,</pre>
22	<pre>    'placeholder', 'none',</pre>
23	<pre>    'imageType', 'CATALOG'</pre>
24	<pre>))"/&gt;</pre>

## cartProductCardBundledItems.isml

Line	storefront_reference_architecture cartridge
8	<code>&lt;img class="product-image"</code>
9	<code>src="\${bundledLineItem.images.small[0].url}"</code>
10	<code>alt="\${bundledLineItem.images.small[0].alt}"</code>
11	<code>title="\${bundledLineItem.images.small[0].title}"</code>
12	<code>&gt;</code>

Line	int_twicpics_sfra cartridge
8	<code>&lt;isinclude url="\${URLUtils.url(</code>
9	<code>    'Twicpics-Image',</code>
10	<code>    'class', 'product-image',</code>
11	<code>    'src', bundledLineItem.images.small[0].url,</code>
12	<code>    'alt', bundledLineItem.images.small[0].alt,</code>
13	<code>    'title', bundledLineItem.images.small[0].title,</code>
14	<code>    'imageType', 'CATALOG'</code>
15	<code>))"/&gt;</code>

## imageAndText.isml

Line	storefront_reference_architecture cartridge
15	<code>&lt;img</code>
16	<code>    class="ITC-image image-fluid common-image-component \${pdict.heading ? 'common-image-filter' : ''}"</code>
17	<code>    src="\${pdict.image.src.mobile}"</code>
18	<code>    style="--focal-point-x:\${pdict.image.focalPointX}; --focal-point-y:\${pdict.image.focalPointY}"</code>
19	<code>    alt="\${pdict.alt ? pdict.alt : imageName}" title="\${pdict.alt ? pdict.alt : imageName}"</code>
20	<code>/&gt;</code>

Line	int_twicpics_sfra cartridge
15	<code>&lt;isinclude url="\${URLUtils.url(</code>
16	<code>    'Twicpics-Image',</code>
17	<code>    'class', pdict.heading ? 'ITC-image image-fluid common-image-component common-image-filter' : 'ITC-image image-fluid common-image-component',</code>
18	<code>    'src', pdict.image.src.mobile,</code>
19	<code>    'alt', pdict.alt ? pdict.alt : imageName,</code>
20	<code>    'title', pdict.alt ? pdict.alt : imageName,</code>
21	<code>    'style', '--focal-point-x:'+pdict.image.focalPointX+'; --focal-point-y:'+pdict.image.focalPointY,</code>
22	<code>    'imageType', 'CONTENT',</code>
23	<code>    'placeholder', 'none'</code>
24	<code>))"/&gt;</code>

## embeddedBonusProduct.isml

Line	storefront_reference_architecture cartridge
11	<code>&lt;img class="product-image"</code>
12	<code>src="\${nestedLineItem.images.small[0].url}"</code>
13	<code>alt="\${nestedLineItem.images.small[0].alt}"</code>
14	<code>title="\${nestedLineItem.images.small[0].title}"</code>
	<code>&gt;</code>

Line	int_twicpics_sfra cartridge
11	<code>&lt;isinclude url="\${URLUtils.url(</code>
12	<code>    'Twicpics-Image',</code>
13	<code>    'class', 'product-image',</code>
14	<code>    'src', nestedLineItem.images.small[0].url,</code>
15	<code>    'alt', nestedLineItem.images.small[0].alt,</code>
16	<code>    'title', nestedLineItem.images.small[0].title,</code>
17	<code>    'imageType', 'CATALOG'</code>
18	<code>))"/&gt;</code>

## productTileSwatch.isml

Line	storefront_reference_architecture cartridge
11	<code>&lt;img class="swatch swatch-circle"</code>
12	<code>data-index="\${loopState.count-1}"</code>
13	<code>style="background-image: url(\${swatch.images.swatch[0].url})"</code>
14	<code>src="\${swatch.images.swatch[0].url}"</code>
15	<code>alt="\${swatch.images.swatch[0].alt}"</code>
	<code>/&gt;</code>

Line	int_twicpics_sfra cartridge
11	<code>&lt;isinclude url="\${URLUtils.url(</code>
12	<code>    'Twicpics-Image',</code>
13	<code>    'class', 'swatch swatch-circle',</code>
14	<code>    'ratio', '100/0',</code>
15	<code>    'placeholder', 'none',</code>
16	<code>    'dataIndex', loopState.count-1,</code>
17	<code>    'style', 'background-image: url('+swatch.images.swatch[0].url+')',</code>
18	<code>    'src', swatch.images.swatch[0].url,</code>
19	<code>    'alt', swatch.images.swatch[0].alt,</code>
20	<code>    'width', '43',</code>
21	<code>    'height', '13',</code>
22	<code>    'imageType', 'CATALOG'</code>
23	<code>))"/&gt;</code>

## imageFile.isml

Line	storefront_reference_architecture cartridge
1	alt="\${pdict.image.alt}"</isif>
6	>

Line	int_twicpics_sfra cartridge
1	<isinclude url="\${URLUtils.url(
2	'Twicpics-Image',
3	'class', 'shopthelook-image image-fluid',
4	'src', pdict.image.url,
5	'alt', pdict.image.alt ? pdict.image.alt : '',
6	'title', product.productName,
7	'imageType', 'CATALOG'
8	)})"/>

## chooseBonusProduct.isml

Line	storefront_reference_architecture cartridge
10	

Line	int_twicpics_sfra cartridge
11	<isinclude url="\${URLUtils.url(
12	'Twicpics-Image',
13	'class', 'product-image',
14	'src', potentialProduct.images.medium[0].url,
15	'alt', potentialProduct.images.medium[0].alt,
16	'title', potentialProduct.images.medium[0].title,
17	'imageType', 'CATALOG'
18	)})"/>

## productCard.isml

Line	storefront_reference_architecture cartridge
17	<code>&lt;img class="product-image"</code>
18	<code>src="\${lineItem.images.small[0].url}"</code>
19	<code>alt="\${lineItem.images.small[0].alt}"</code>
20	<code>title="\${lineItem.images.small[0].title}"</code>
	<code>&gt;</code>

Line	int_twicpics_sfra cartridge
17	<code>&lt;isinclude url="\${URLUtils.url(</code>
18	<code>    'Twicpics-Image',</code>
19	<code>    'class', 'product-image',</code>
20	<code>    'src', lineItem.images.small[0].url,</code>
21	<code>    'alt', lineItem.images.small[0].alt,</code>
22	<code>    'title', lineItem.images.small[0].title,</code>
23	<code>    'imageType', 'CATALOG'</code>
24	<code>))"/&gt;</code>

## uncategorizedCartProductCard.isml

Line	storefront_reference_architecture cartridge
15	<code>&lt;img</code>
	<code>    class="product-image"</code>
	<code>src="\${lineItem.images.small[0].url}"</code>
	<code>alt="\${lineItem.images.small[0].alt}"</code>
	<code>title="\${lineItem.images.small[0].title}"</code>
	<code>&gt;</code>

Line	int_twicpics_sfra cartridge
15	<code>&lt;isinclude url="\${URLUtils.url(</code>
16	<code>    'Twicpics-Image',</code>
17	<code>    'class', 'product-image',</code>
18	<code>    'src', lineItem.images.small[0].url,</code>
19	<code>    'alt', lineItem.images.small[0].alt,</code>
20	<code>    'title', lineItem.images.small[0].title,</code>
21	<code>    'imageType', 'CATALOG'</code>
22	<code>))"/&gt;</code>

## bonusProductCard.isml

Line	storefront_reference_architecture cartridge
17	

Line	int_twicpics_sfra cartridge
17	<isinclude url="\${URLUtils.url(
18	'Twicpics-Image',
19	'class', 'product-image',
20	'src', lineItem.images.small[0].url,
21	'alt', lineItem.images.small[0].alt,
22	'title', lineItem.images.small[0].title,
23	'imageType', 'CATALOG'
24	)}"/>

## productTileImage.isml

Line	storefront_reference_architecture cartridge
3	

Line	int_twicpics_sfra cartridge
3	<isinclude url="\${URLUtils.url(
4	'Twicpics-Image',
5	'class', 'tile-image',
6	'src', product.images.medium[0].url,
7	'alt', product.productName,
8	'title', product.images.medium[0].title,
9	'imageType', 'CATALOG'
10	)}"/>



## productShippingCard.isml

Line	storefront_reference_architecture cartridge
13	<pre>&lt;img   class="product-image"   src="\${lineItem.images.small[0].url}"   alt="\${lineItem.images.small[0].alt}"   title="\${lineItem.images.small[0].title}" &gt;</pre>
14	

Line	int_twicpics_sfra cartridge
13	<pre>&lt;isinclude url="\${URLUtils.url(   'Twicpics-Image',   'class', 'product-image',   'src', lineItem.images.small[0].url,   'alt', lineItem.images.small[0].alt,   'title', lineItem.images.small[0].title,   'imageType', 'CATALOG' )})"/&gt;</pre>
14	
15	
16	
17	
18	
19	
20	

## productTile.isml

Line	storefront_reference_architecture cartridge
15	<pre>&lt;img class="product-tile-component-image"   src="\${product.images.medium[0].url}"   alt="\${product.productName}"   title="\${product.images.medium[0].title}" &gt;</pre>
16	
17	
18	
19	

Line	int_twicpics_sfra cartridge
15	<pre>&lt;isinclude url="\${URLUtils.url(   'Twicpics-Image',   'class', 'product-tile-component-image',   'src', product.images.medium[0].url,   'alt', product.productName,   'title', product.images.medium[0].title,   'imageType', 'CATALOG' )})"/&gt;</pre>
16	
17	
18	
19	
20	
21	
22	

## mainBanner.isml

Line	storefront_reference_architecture cartridge
14	<code>&lt;img</code>
15	<code>class="mainbanner-image image-fluid common-image-component common-image-filter"</code>
16	<code>src="\${pdict.image.src.mobile}"</code>
17	<code>style="--focal-point-x:\${pdict.image.focalPointX}; --focal-point-</code>
18	<code>y:\${pdict.image.focalPointY}"</code>
18	<code>&lt;isif condition="\${pdict.image.alt}"&gt;alt="\${pdict.image.alt}" title="\${pdict.image.alt</code>
19	<code>}"&lt;/isif&gt;</code>
19	<code>/&gt;</code>

Line	int_twicpics_sfra cartridge
14	<code>&lt;isinclude url="\${URLUtils.url(</code>
15	<code>    'Twicpics-Image',</code>
16	<code>    'class', 'mainbanner-image image-fluid common-image-</code>
17	<code>component common-image-filter',</code>
17	<code>    'src', pdict.image.src.mobile,</code>
18	<code>    'alt', pdict.image.alt ? pdict.image.alt : '',</code>
19	<code>    'title', pdict.image.alt ? pdict.image.alt : '',</code>
20	<code>    'style', '--focal-point-x:'+pdict.image.focalPointX+'; --</code>
21	<code>focal-point-y:'+pdict.image.focalPointY,</code>
22	<code>    'placeholder', 'none',</code>
23	<code>    'imageType', 'CONTENT',</code>
24	<code>    'width', '1106',</code>
25	<code>    'height', '556'</code>
25	<code>)}"/&gt;</code>

## categoryTile.isml

Line	storefront_reference_architecture cartridge
18	<code>&lt;img</code>
	<code>class="img-fluid"</code>
	<code>src="\${category.custom.slotBannerImage.getURL()}"</code>
	<code>alt="\${category.displayName}"</code>
	<code>/&gt;</code>

Line	int_twicpics_sfra cartridge
18	<code>&lt;isinclude url="\${URLUtils.url(</code>
19	<code>    'Twicpics-Image',</code>
20	<code>    'class', 'img-fluid',</code>
21	<code>    'src', category.custom.slotBannerImage.getURL(),</code>
22	<code>    'alt', category.displayName,</code>
23	<code>    'imageType', 'CATALOG')}"&gt;</code>

## suggestions.isml

Line	storefront_reference_architecture cartridge
33	<pre>&lt;img   class="swatch-circle hidden-xs-down"   alt="\${product.name}"   src="\${product.imageUrl}" /&gt;</pre>
55	<pre>&lt;img   class="swatch-circle hidden-xs-down"   alt="\${category.name}"   src="\${category.imageUrl}" /&gt;</pre>

Line	int_twicpics_sfra cartridge
33	<pre>&lt;isinclude url="\${URLUtils.url(</pre>
34	<pre>  'Twicpics-Image',</pre>
35	<pre>  'class', 'swatch-circle hidden-xs-down',</pre>
36	<pre>  'src', product.imageUrl,</pre>
37	<pre>  'alt', product.name,</pre>
38	<pre>  'imageType', 'CATALOG'</pre>
39	<pre>)}"/&gt;</pre>
40	
62	<pre>&lt;isinclude url="\${URLUtils.url(</pre>
63	<pre>  'Twicpics-Image',</pre>
64	<pre>  'class', 'swatch-circle hidden-xs-down',</pre>
65	<pre>  'src', category.imageUrl,</pre>
66	<pre>  'alt', category.name,</pre>
67	<pre>  'imageType', 'CATALOG'</pre>
68	<pre>)}"/&gt;</pre>
69	

## 9. Operations, Maintenance

### Availability

<EXPECTED AVAILABILITY /UPTIME OF ANY EXTERNAL SERVICE, INTERFACES>  
<FALLBACK SOLUTION, BEHAVIOR IF EXTERNAL SERVICES ARE NOT AVAILABLE, IMPACT ON CUSTOMER STOREFRONT>  
<ANY EXISTING UTILITIES THAT HELP TO DETECT AVAILABILITY/UPTIME OF EXTERNAL SERVICE, E.G. WEBSERVICE CALL, GOMEZ PING>  
<ESTIMATED PERFORMANCE METRICS FOR PEAK BUSINESS HOURS IF AVAILABLE>  
<NOTIFICATION PROCESS IF EXTERNAL SERVICES, INTERFACES ARE NOT RESPONDING, E.G. HOTLINE /SUPPORT PHONE NUMBER>

### Failover/Recovery Process

<WHAT MERCHANTS SHOULD EXPECT WHEN SERVICE IS DOWN AND WHAT TO DO>

### Support

<CONTACT EMAIL IN CASE DEFECT FIXES OR IMPROVEMENTS FOR COMPONENT ARE REQUIRED>

## 10. Release history

Version	Date	Changes
21.1.0	16/06/2021	Initial release