



Media Publisher Delivery Component (MDC)

Thu Nov 27 15:46:40 UTC 2025

## Table of Contents

How It Works .....	2
Delivery Mechanism .....	2
Original Assets .....	2
Image Renditions .....	2
Publishing Modes .....	2
On-Demand-Publishing .....	2
Manual Publishing .....	2
Delivering Cropped Images .....	3
Focus Area Cropping .....	3
Format of a Focus Area Crop Template .....	3
URL Format .....	5
Asset URL Format .....	5
Examples: .....	5
Image Rendition URL Format .....	5
Image Rendering Operations .....	6
Customer-Owned Resources .....	10
S3 Buckets .....	10
CloudFront .....	10
Origin Groups in CloudFront .....	10
Lambda@Edge for MDC .....	10
Support for Custom Download Names .....	11
Support for Content-Disposition .....	11
Redirect to Fallback Image on Error .....	12
Chinese CloudFront Distribution .....	12
S3 Bucket Configuration for China .....	12

[Show in Fullscreen](#)

[Download as PDF](#)

Users of the Canto DAM system want to be able to publish images and documents worldwide. Media Publisher Delivery Component (MDC) is an option providing fast delivery of images and documents as a cloud-based service.

## How It Works

MDC can publish a Canto asset in one or both of the following ways:

- As an original asset which is then available for download as-is
- As a master image which then allows requesting different renditions that are generated on demand from the master image.

The user can decide to publish an asset as an original or as a master image to let MDC deliver different renditions on the fly. The user can also decide to publish an asset in both ways if the original image should also be available for download.

MDC is running on Amazon WebServices (AWS), a cloud platform that is widely used and tested and consists of several components.

## Delivery Mechanism

### Original Assets

To deliver original assets the CloudFront Distribution firsts calls the MDC web service to make sure the asset is available in the corresponding S3 bucket. If the asset is not yet available it is copied from the Canto DAM system to the S3 bucket. When delivering the asset the CloudFront distribution delivers it directly from the S3 storage, caching the asset in the local edge node closest to the user requesting it so that any subsequent request for the same asset hitting the same edge node is delivered from cache.

### Image Renditions

To deliver image renditions the CloudFront Distribution firsts calls the MDC web service to make sure the rendition is available in the corresponding S3 bucket. If the rendition is not yet available the MDC web service will render the image and store it in the S3 bucket before returning the call. The web service responds by redirecting CloudFront to another CloudFront URL which then delivers the rendition from the S3 bucket directly.

Renditions of the master image can include the following processing steps:

- Scale the image by a given factor, to a target width and height, or to a target bounding box
- Crop a certain rectangular region out of the image
- Specify the output file format

MDC optionally delivers a fallback image in case of problems with the original image

### Publishing Modes

MDC support two different modes of operation: on-demand and manual publishing. The mode is configured on a customer-by-customer base.

### On-Demand-Publishing

In this mode each asset or rendition is published and MDC retrieves the required assets or master image from Canto DAM as they are requested through CloudFront. Assets, master images, and renditions are purged automatically 6 months after they have been created. They are re-created on demand.

### Manual Publishing

In this mode an explicit request sent to the MDC service creates the asset or master image in MDC which is then available for delivery. Any request for an asset or rendition where the asset or master image is not already available in MDC is rejected and MDC responds with an HTTP status 404 (Not Found). In this mode the assets, master images, and renditions are not automatically deleted from MDC once they have expired. Instead, a separate explicit request sent to MDC allows removing the asset or image from MDC, so they are no longer published.

Delivering an Original Asset The following diagrams show the different requests being made to deliver an original asset depending on whether the asset was found in the CloudFront cache.

### Delivering Cropped Images

MDC supports two ways of cropping images:

- Focus Area Cropping
- Manual Cropping

#### Focus Area Cropping

With focus area cropping the administrator sets up the crop template with the resulting dimensions. Focus area crop templates are stored in the MDC account item in DynamoDB. The user only specifies a "region of interest" (ROI) in the image which should be preserved when cropping. The ROI is stored for each image in MDC in a separate table. If the ROI is missing for an image cropping using a focus area crop template uses the whole image as the "region of interest."

To get a cropped image the rendition URL only contains a "-T" rendition parameter with the crop template name or the URL contains a "-R" rendition parameter with the crop template values. MDC itself does the rest when delivering the cropped image.

#### Format of a Focus Area Crop Template

The MDC account item in the database of MDC may contain a property metadata containing cropTemplates which defines the focus area crop templates for this customer.

#### Focus Area Crop Template Configuration

```
{
  "id": "123456789",
  "name": "Acme Corp.",
  "metadata": {
    "cropTemplates": [
      {
        "name": "banner",
        "orientation": 5,
        "size": {
          "w": 1200,
```

```
    "h": 50
  }
},
{
  "name": "vertical",
  "orientation": 1,
  "size": {
    "w": 200,
    "h": 1500
  }
}
]
}
```

The orientation specifies where in the resulting image the region of interest should appear. A value of 0 indicates no specific location.

Otherwise, the orientation can be one of the following values:

Table 1. Crop Template "orientation" values

1 = top left	2 = top center	3 = top right
4 = left center	5 = center	6 = right center
7 = bottom left	8 = bottom center	9 = bottom right

The size specifies the desired output size in pixels.

## URL Format

### Asset URL Format

A URL for an original asset should always be generated by the MDC service. It will then look like the following:

***Scheme://CloudFrontDomain/asset/Account\_ID/Asset\_ID/DownloadName***

Table 2. Asset URL Format

Placeholder	Description
Scheme	The protocol (http or https)
CloudFrontDomain	The CloudFront domain name (e.g. d5ek9eu3ld9fw.cloudfront.net)
Account_ID	The MDC account ID, typically the AWS account ID of the customer
Asset_ID	The identifier of the asset in MDC. This is the Canto scheme and the Canto content ID separated by underline, e.g. document_3ck4lkvbk134h0i9hsbs9gk229
DownloadName	(Optional) A download name to be given to the delivered asset. When the user downloads the asset using a browser it will offer this name when saving to disk.

Optionally a query parameter named "content-disposition" with a value of "inline" can be added to the uRL to let the browser render the asset in the web page instead of downloading it to disk.

### Examples:

<https://d24tfwkmonftaj.cloudfront.net/asset/123456789/>

[document\\_3ck4lkvbk134h0i9hsbs9gk229](#)

Download the asset with the MDC asset ID document\_3ck4lkvbk134h0i9hsbs9gk229 using the default download name (document\_3ck4lkvbk134h0i9hsbs9gk229)

<https://d24tfwkmonftaj.cloudfront.net/asset/123456789/>

[document\\_3ck4lkvbk134h0i9hsbs9gk229/manual\\_en\\_1234.pdf](#)

Download the asset (a PDF document) with the MDC asset ID document\_3ck4lkvbk134h0i9hsbs9gk229 using the download name manual\_en\_1234.pdf.

<https://d24tfwkmonftaj.cloudfront.net/asset/123456789/>

[document\\_3ck4lkvbk134h0i9hsbs9gk229?content-disposition=inline](#)

Display the asset (e.g. a PDF document) with the MDC asset ID document\_3ck4lkvbk134h0i9hsbs9gk229 in the current browser window. If the browser does not support rendering the file format it will download the asset.

### Image Rendition URL Format

A URL for an image rendition should always be generated by the MDC service. It will then look like the following:

***Scheme://CloudFrontDomain/image/Account\_ID/Image\_ID/Operations***

For legacy customers the /image/ part needs to be replaced by /rendition/ in the URL.

Table 3. Image Rendition URL Format

Placeholder	Description
Scheme	The protocol (http or https)
CloudFrontDomain	The CloudFront domain name (e.g. d5ek9eu3ld9fw.cloudfront.net)
Account_ID	The MDC account ID, typically the AWS account ID of the customer
Image_ID	The identifier of the image in MDC. This is the Canto scheme and the Canto content ID separated by underline, e.g. image_3ck4lkvbk134h0i9hsbs9gk229
Operations	(Optional) The image processing parameters for converting the master image into the rendition. See the table below for a list of image operations.

### Image Rendering Operations

This part of the image rendition URL specifies which image operations should be applied to the master image. If this is missing in the URL MDC will use the default rendition operations configured in the MDC account. You can specify multiple operations which are processed in their order of appearance in the URL, e.g. -C50%x50%,25%,25%-S200x200-FPNG will first crop an area of 50% of the image size at the center of the image and then scale it down to fit an area of 200 by 200 pixels. The resulting image is delivered in PNG format.

For example, the operations -C50%x50%,25%,25%-S200x200-FPNG will be applied in the following order: . -C50%x50%,25%,25% Crops an area with 50% of the master images width and height, with an offset from the top and left of 25% of the master images width and height . -S200x200 Scales the area cropped by the previous operation to an absolute width and height of 200 pixels . -FPNG Renders the result of the previous two operations in the PNG format

Table 4. Image Rendition Operations

Operation	Description
- <b>widthxheight</b>	Scale the image to a given <b>width</b> and <b>height</b> specified in pixels. It will scale up or down depending on the master image and the requested size.  Example: -S500x200 Scale to 500 x 200 pixels size.
- <b>size</b>	Scale the image to fit into a square box of the given <b>size</b> in pixels. It will scale up or down depending on the master image and the requested size.  Example: -B300 Scale to fit into 300x300 pixels

Operation	Description
- <b><i>widthxheight</i></b>	<p>Crop a rectangle of the specified <b><i>width</i></b> and <b><i>height</i></b> from the top-left of the image.</p> <p>Example: -C200x200 Crop an area of 200 x 200 pixels</p> <p><b>Please Note:</b> The <b><i>width</i></b> and <b><i>height</i></b> values used in a cropping operation should not exceed the width and height of the master image. If a provided size parameter exceeds the maximum size of the master image, that parameter will default to the master image's maximum size instead.</p> <p>For example, for a master image with a width and height of 1000 x 1000 pixels:</p> <p>The operation -C2000x500 will result in a cropped area of 1000 x 500 pixels</p> <p>The operation -C500x2000 will result in a cropped area of 500 x 1000 pixels</p> <p>The operation -C2000x2000 will result in a cropped area of 1000 x 1000 pixels</p> <p>* <b>Note:</b> All operation which will follow up will work on the returned image size and not on the requested image size.*</p>
- <b><i>widthxheight,x,y</i></b>	<p>Crop a rectangle of the specified <b><i>width</i></b> and <b><i>height</i></b> of the image at the given x and y offsets from left and top respectively.</p> <p>All crop values can either be specified in pixels or as a value relative to the image size as a percentage value.</p> <p>Example: -C20%x20%,15.32%,2.45% Crop an area of 20% of the total size at the position 15.32% from the left and 2.45% from the top.</p>

Operation	Description
- <b>TtemplateID</b>	Crop according to the crop template of the given <b>templateID</b> . If the crop template is a focus area crop template MDC will automatically crop according to the "Region Of Interest" value of the given image. If the ROI is missing it will take the whole image as the region of interest.
- <b>Fformat,Quality</b>	Deliver the rendition image in the given file format using the given quality. Possible <b>format</b> values are JPG, PNG, TIF, GIF, WEBP, and JP2 default is PNG. The <b>quality</b> option is supported only for "JPG" file format and has a range from 1 to 100, default is 85.
- <b>WwatermarkID</b>	Render a watermark into the image using a 50% transparency. The watermark image is marked specifically in MDC.
- <b>RtemplateValues</b>	Crop according to the focus area crop <b>templateValues</b> . MDC automatically crops according to the "Region Of Interest" value of the given image. If the ROI is missing it will take the whole image as the region of interest.  Example: -Ro:7,w:100,h:200,n:default

Table 5. Focus area crop template values

Option	Description	
o: <b>orientation</b>	The <b>orientation</b> specifies where in the resulting image the region of interest should appear. A value of 0 indicates no specific location. Otherwise, the <b>orientation</b> can be one of the following values:	
	Orientation value	Description
	1, top_left or tl	top left
	2, top_center or tc	top center
	3, top_right or tr	top right
	4, middle_left or ml	left center
	5, middle_center or mc	center
	6, middle_right or mr	right center
	7, bottom_left or bl	bottom left
	8, bottom_center or bc	bottom center
9, bottom_right or br	bottom right	
n: <b>regionOfInterest</b>	Identifier of the region of interest. If not specified, the region of interest with the identifier "default" is used.	

Option	Description
w: <b>width</b>	The <b>width</b> specifies the desired output width in pixels.
h: <b>height</b>	The <b>height</b> specifies the desired output height in pixels.

## Customer-Owned Resources

MDC works with some AWS resources that belong to the customer's AWS account: the S3 buckets, the CloudFront distribution, and the Lambda@Edge function used by the CloudFront distribution.

### S3 Buckets

The S3 buckets that belong to the customer's AWS account store the MDC asset, master image, and rendition image objects. The MDC service manages these objects automatically. There is no need to manipulate them manually, actually this may cause problems because every object in S3 is referenced by a database item inside the MDC service and MDC keeps them consistent.

### CloudFront

The delivery of assets and images is always using the Content Delivery Network service of AWS called "CloudFront." The purpose of CloudFront is to cache results locally for faster delivery to the end user. But CloudFront also has some logic that helps to implement some MDC features. A CloudFront distribution is an HTTPS endpoint that can deliver original assets as well as renditions of images. The configuration of a distribution includes mapping a URL path to a specific origin that CloudFront uses in case the resource is not found in its cache.

### Origin Groups in CloudFront

The configuration uses one origin group for assets and another group for renditions. An origin group contains a list of origins where the second origin is only used in case the first one returns an error. The first origin is the S3 bucket for assets or renditions respectively. The second origin is the MDC service endpoint that obtains the requested resource and stores it at the S3 bucket and then responds with a redirect status redirecting to the same URL. This way CloudFront first tries to directly access the requested resource and only calls the MDC service if that resource has to be generated improving the overall performance.

### Lambda@Edge for MDC

Another way to add logic to a CloudFront distribution is to configure a Lambda function to be executed inside the CloudFront edge node (the technology is called "Lambda@Edge"). The benefit is that this lambda function is executed close to the end user and can control the request sent to the MDC service as well as the exact response to the user's browser.

The Lambda@Edge function for MDC has the following purposes:

- Add support for custom download names for original assets and image renditions
- Add support for content-disposition for original assets
- Redirect to the fallback image URL in case the MDC service returns an error for rendering an image
- Work around a bug in CloudFront and API Gateway when handling "%" characters in URLs

The MDC Lambda@Edge function should be configured for all events that CloudFront supports:

- Origin Response
- Origin Request
- Viewer Request
- Viewer Response

### **Support for Custom Download Names**

When requesting an original asset or rendition, the URL ends with the Asset ID (e.g. "document\_9avcaun14d1pl23skf04dpks22") of the asset or the rendition parameters (e.g. "-FPNG") which will become the file name suggested by the browser to store the asset or rendition in a file. Adding a custom name to the end of the CloudFront URL separated by "/" allows to specify a custom download name.

In the CloudFront Distribution the Lambda@Edge function checks whether the URL coming from the browser contains such a custom name. It then removes this name from the request URI and remembers it for later use. This way the URL without the name is sent to the origin (using the same origin request for all custom names) and the CloudFront cache also only caches the URL without the custom name.

In case the response redirects to a different URL (e.g. when requesting a /rendition which redirects to /image) the custom download name is added to the redirect location URI, so it is preserved for the following request by the browser.

When the response is later sent back to the browser it will suggest the custom name as the file name to save the asset because it was the last part of the URL that the browser requested.

### **Support for Content-Disposition**

When requesting an original asset, MDC by default returns a Content-Disposition with the value of "attachment" causing the browser to download the original asset instead of displaying it in a web page. By specifying a query parameter "content-disposition" with the value of "inline" in the URL the asset will be displayed in the surrounding web page.

When requesting a rendition MDC, by default returns a Content-Disposition with the value of "inline" causing the browser to display the rendition in a web page instead of downloading it. By specifying a query parameter "content-disposition" with the value of "attachment" in the URL the rendition will be downloaded.

In the CloudFront Distribution the Lambda@Edge function reacts depending on whether the request was successfully satisfied by the S3 origin or by the MDC service.

If the S3 origin returns a success (HTTP Status 200) the Lambda@Edge function sets the HTTP header "Content-Disposition" according to the value of the query parameter (or to "attachment" in case it is missing).

If the MDC service origin returns a success (HTTP Status 302) the Lambda@Edge function adds the "content-disposition" query parameter to the redirect location URL to request the same

content disposition information with the redirect URL.

This feature can be combined with the custom name feature described above to display PDF documents inline but use a custom file name when downloading the PDF to a file.

### **Redirect to Fallback Image on Error**

In the CloudFront Distribution the Lambda@Edge function reacts on errors returned by the MDC service (which is the fallback origin for the S3 origin).

If the MDC service returns an error on generating an image rendition the Lambda@Edge function returns an HTTP status 302 (Redirect) with a /rendition/ URL containing the virtual image ID "fallbackimage" instead of the image ID requested. This is returned to the browser causing another request directly to the MDC service (because an image with ID "fallbackimage" never exists in the S3 origin). The MDC service then will generate a rendition based on the configured fallback image and will respond with another HTTP status 302 to the /image/ URL of the fallback image. The Lambda@Edge instructs CloudFront and the browser not to cache the redirection to the fallbackimage response because it should only be a temporary fix for a problem (e.g. an invalid image ID).

### **Chinese CloudFront Distribution**

Then a separate CloudFront Distribution needs to be created in an AWS China region. Because MDC then has to deal with two different CloudFront locations a single base URL configured in the MDC account is not sufficient. To allow redirecting to the correct CloudFront distribution MDC supports a custom HTTP header when handling requests from CloudFront. The "Canto-Redirect-Base" header can contain the base URL of the requesting CloudFront instance to let MDC redirect to this instead of the configured base URL.

When configuring the CloudFront distribution in China the origin for the MDC service should be configured to contain a custom header named "Canto-Redirect-Base" with the base URL of the current CloudFront distribution as the value (see the "cloudFrontDomain" property of the MDC account configuration below).

### **S3 Bucket Configuration for China**

While a CloudFront distribution in China can access the MDC endpoint the access to the S3 buckets only works if the buckets are configured to act as a static website.

Since the S3 bucket for the master images are not accessed by CloudFront only the S3 buckets for assets and for rendition need to be changed.

In the S3 buckets for assets (canto-mdc-assets-XXXXXXXX) and for renditions (canto-mdc-renditions-XXXXXXXX) enable the "Static website hosting."