

Serving Direct Update requests from a CDN

Overview

You can configure Direct Update requests to be served from a CDN (content delivery network) instead of from the MobileFirst Server.

Advantages of using a CDN

Using a CDN instead of the MobileFirst Server to serve Direct Update requests has the following advantages:

- Removes network overheads from the MobileFirst Server.
- Increases transfer rates higher than the 250 MB/second limit when serving requests from a MobileFirst Server.
- Ensures a more uniform Direct Update experience for all users regardless of their geographical location.

General requirements

To serve Direct Update requests from a CDN, ensure that your configuration conforms to the following conditions:

- The CDN must be a reverse proxy in front of the MobileFirst Server (or in front of another reverse proxy if needed).
- When building the application from your development environment, set up your target server to the CDN host and port instead of the host and port of the MobileFirst Server. For example, when running the MobileFirst Platform CLI command `mpfdev server add`, provide the CDN host and port.
- In the CDN administration panel, you need to mark the following Direct Update URLs for caching to ensure that the CDN passes all requests to the MobileFirst Server except for the Direct Update requests. For Direct Update requests, the CDN determines whether it obtained the content. If it has, it returns it without going to the MobileFirst Server; if not, it goes to the MobileFirst Server, gets the Direct Update archive (.zip file), and stores it for the next requests for that specific URL. For applications that are built with v8.0 of IBM MobileFirst Foundation, the Direct Update URL is: `PROTOCOL://DOMAIN:PORT/CONTEXT_PATH/api/directupdate/VERSION/CHECKSUM/TYPE`. The `PROTOCOL://DOMAIN:PORT/CONTEXT_PATH` prefix is constant for all runtime requests. For example:
`http://my.cdn.com:9080/mpf/api/directupdate/0.0.1/742914155/full?appld=com.ibm.DirectUpdateTestApp&clientPlatform=android`

In the example, there are additional request parameters that are also part of the request.

- The CDN must allow caching of the request parameters. Two different Direct Update archives might differ only by the request parameters.
- The CDN must support TTL on the Direct Update response. The support is needed to support multiple direct updates for the same version.
- The CDN must not change or remove the HTTP headers that are used in the MobileFirst server-client protocol.

Example configuration

This example is based on using an Akamai CDN configuration that caches the Direct Update archive. The following tasks are completed by the network administrator, the MobileFirst administrator, and the Akamai administrator:

Network administrator

Create another domain in the DNS for your MobileFirst Server. For example, if your server domain is `yourcompany.com` you need to create an additional domain such as `cdn.yourcompany.com`. In the DNS for the new `cdn.yourcompany.com` domain, set a **CNAME** to the domain name that is provided by Akamai. For example, `yourcompany.com.akamai.net`.

MobileFirst administrator

Set the new `cdn.yourcompany.com` domain as the MobileFirst Server URL for the MobileFirst applications. For example, for the Ant builder task, the property is: `<property name="wl.server" value="http://cdn.yourcompany.com/${contextPath}"/>`.

Akamai administrator

1. Open the Akamai property manager and set the property **host name** to the value of the new domain.

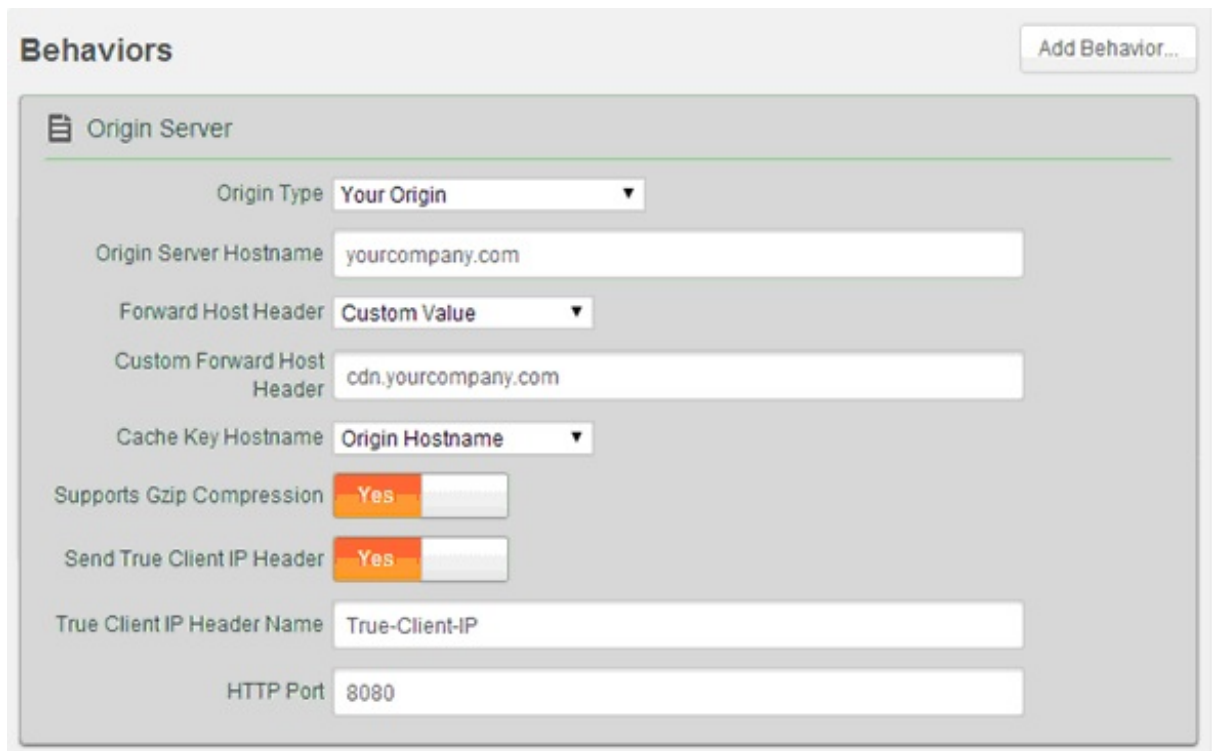


PROPERTY HOSTNAMES 


HOSTNAME	EDGE HOSTNAME
<input type="text" value="cdn.yourcompany.com"/>	<input type="text" value="yourcompany.com.akamai.net"/> 



2. On the Default Rule tab, configure the original MobileFirst Server host and port, and set the **Custom Forward Host Header** value to the newly created domain.



Behaviors

 **Origin Server**

Origin Type

Origin Server Hostname

Forward Host Header

Custom Forward Host Header

Cache Key Hostname

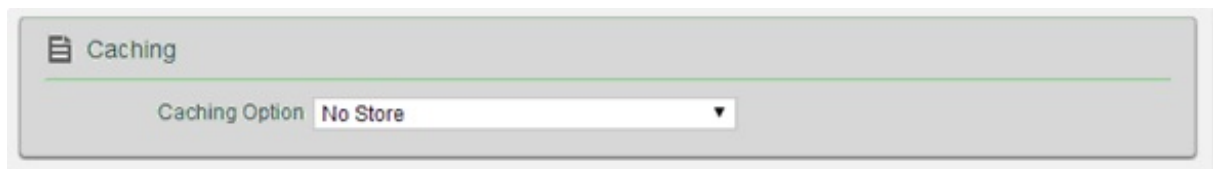
Supports Gzip Compression ☒ Yes

Send True Client IP Header ☒ Yes

True Client IP Header Name

HTTP Port


3. From the **Caching Option** list, select **No Store**.



Caching

Caching Option **No Store**

- From the **Static Content configuration** tab, configure the matching criteria according to the Direct Update URL of the application. For example, create a condition that states **If Path matches one of direct_update_URL**.



Criteria **Match All** **Add Match**

IF

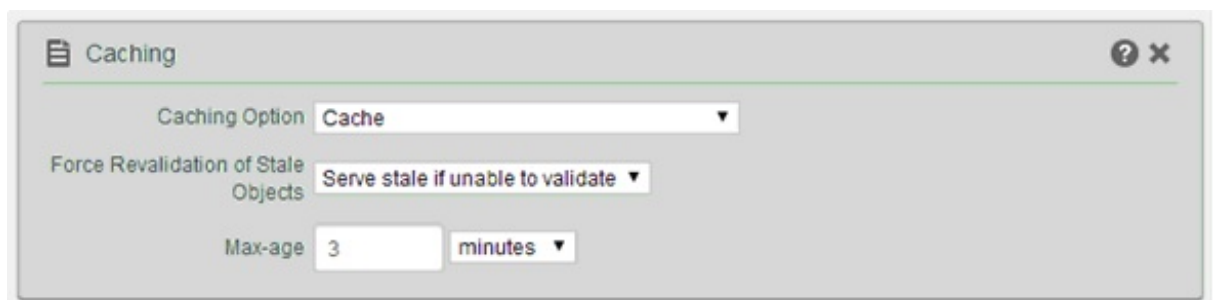
Path

matches one of

/WorklightStarter/apps/services/api/WorklightStarter/updates

- Set values similar to the following values to configure the caching behavior to make cache the Direct Update URL and to set TTL.

Field	Value
Caching Option	Cache
Force Revaluation of Stale Objects	Serve stale if unable to validate
Max-Age	3 minutes



Caching

Caching Option **Cache**

Force Revaluation of Stale Objects **Serve stale if unable to validate**

Max-age **3** **minutes**

- Configure the cache key behavior to use all request parameters in the cache key (you must do so to cache different Direct Update archives for different applications or versions). For example, from the **Behavior** list, select **Include all parameters (preserve order from request)**.



Cache Key Query Parameters

Behavior **Include all parameters (preserve order from request)**