Citi China CMS:
Manual Cloudflare Intervention

Interim Reference Guide — Akamai Exit

Scope: siteId = `citi-china`
Date: June 2026
Context: Citi China Akamai → Cloudflare CDN Migration
Audience: CMS Administrators, DevOps, Platform Support

INTERIM — MANUAL STEPS REQUIRED UNTIL CLOUDFLARE API INTEGRATION

Overview: What Changed and Why Manual Intervention Is Needed

What is happening?

Citi China (`siteId=citi-china`) is exiting Akamai as its CDN provider and migrating to **Cloudflare**. The CMS platform (retail-service-integration, citicms-service, retail-content-service, and citicms-ui) has deep Akamai API integration that automates CDN cache purging, asset uploads to Akamai NetStorage (AKNS), and data synchronisation between MongoDB and the CDN layer.

When Akamai is decommissioned for Citi China, all of these automated integrations will stop functioning for the `citi-china` siteId. The Cloudflare CDN will be in place to serve content, but the CMS backend has **no Cloudflare API integration yet**.

Why do we need to do this manually?

The CMS codebase currently has a pluggable `cacheProvider` architecture (see `AbstractFastPurgeService` and `SSGConfigurationChecker.getCacheProvider()`), but only an Akamai implementation exists today (`AkamaiFastPurgeService`). A Cloudflare implementation has not yet been built. Until the team:

1. Implements a `CloudflareFastPurgeService` extending `AbstractFastPurgeService`
2. Adds Cloudflare R2/Workers equivalents for Akamai NetStorage upload/delete operations
3. Updates SiteSetting for `citi-china` to set `cacheProvider=cloudflare`
4. Deploys these changes across all backend services

...every CDN operation for Citi China must be performed manually via the Cloudflare dashboard or CLI.

Citi China Configuration

Parameter	Value
siteId	`citi-china`
Asset Prefix	`/chn/chn-ui/akpublic`
Frontend Port (public)	5400
Frontend Port (preview)	5450
Frontend Server	`citiCountryUiServer.js`
SSG (frontend flag)	`false` (server-rendered; SSG engine may still be enabled via SiteSetting)
CKEditor Styles	Custom styling enabled (alongside citigpa, citi-idn)

How Akamai integration works in the CMS

The CMS backend services drive Akamai through these key mechanisms, all checked per `siteId` via SiteSetting:

- **Cache Purge (Fast Purge)** — `AkamaiFastPurgeService` invalidates/deletes URLs from Akamai edge cache. Triggered by page publish/withdraw, SSG regeneration, and content updates. Controlled by `cacheProvider=akamai`

in SiteSetting > SSG.

- **NetStorage (AKNS)** — `AkamaiNSFeedService` uploads/deletes images to Akamai NetStorage origin. Controlled by `enableAKNS=true` in SiteSetting > Assets.
- **CDN Content Query Feed** — `AkamaiFeedContentQueryService` purges JSON API cache for CDN-enabled content types. Controlled by `enableQueryFromCDN=true` per content definition.
- **Asset Link Purge** — `AssetAkamaiServiceImpl` purges CDN cache for asset share links after TTS PDF generation or content sync. Domain-aware with staging/public purge support.

Table of Contents

1. Publish a Page (Page Builder) IF SSG ENABLED

CMS Steps:

1. User edits a `citi-china` page in Page Builder
2. User clicks **Publish**
3. CMS saves the page to DB and sends a Kafka message (`EVENT_PUBLISH` , resource type `page`)
4. If SSG is enabled for `citi-china` , the SSG engine generates static HTML
5. `AbstractSsgKafkaConsumer` calls `AkamaiFastPurgeService.purgeCache()` to invalidate the page URL on CDN BROKEN

What breaks: The published page URL continues to serve the old cached version on the Citi China public domain. With `Edge-Control: max-age=7d` TTL, stale content may persist for up to 7 days.

Manual Cloudflare action: Log in to the Cloudflare dashboard for the Citi China zone and purge the specific page URL. The URL is the page's public URL on the `citi-china` domain.

Applicability note: The frontend startup script sets `ssg=false` for `citi-china` . If SSG is also disabled in SiteSetting (`enableSsg=false`), pages are rendered server-side and this workflow does not apply. However, CDN may still cache server-rendered pages, so cache purging may still be needed depending on Cloudflare cache rules.

Source: `AbstractSsgKafkaConsumer.java:148,249` , `AkamaiFastPurgeService.java:38-55` , `SSGConfigurationChecker.java:106-108`

2. Withdraw a Page IF SSG ENABLED

CMS Steps:

1. User opens a published `citi-china` page in Page Builder
2. User clicks **Withdraw**
3. CMS updates the page status to `DRAFT` and sends a Kafka message (`EVENT_WITHDRAW` , resource type `page`)
4. SSG engine deletes the static HTML file
5. `AkamaiFastPurgeService` deletes the page URL from CDN cache BROKEN

What breaks: The withdrawn page remains accessible via Cloudflare CDN for up to 7 days. Users visiting the public URL will still see the content that should have been removed.

Manual Cloudflare action: Purge the specific page URL from Cloudflare cache immediately after withdrawing.

Source: `AkamaiFastPurgeService.java:112-115`

3. Publish Content (CDN-Enabled Content Types) IF enableQueryFromCDN

CMS Steps:

1. User edits a `citi-china` content item (e.g., FAQ, userGuide, or any type with `enableQueryFromCDN=true`)
2. User clicks **Publish**
3. CMS saves the content and sends a Kafka message (`EVENT_PUBLISH` , resource type `content`)
4. `AkamaiFeedContentQueryService` checks if the content type has `enableQueryFromCDN=true` in its content definition for `citi-china` , then builds CDN URLs and purges them BROKEN

What breaks: The CDN API endpoints continue serving stale JSON. The URL patterns for `citi-china` are:
`{domain}/rcs/v1/akamaiCDN/citi-china/content/{type}/businessId/{businessId}.json`
`{domain}/rcs/v1/akamaiCDN/citi-china/content/{type}/id/{id}.json`
`{domain}/rcs/v2/siteIds/citi-china/akamaiCDN/contents/{type}/businessId/{businessId}.json?language={lang}` (if multi-language enabled)

For TTS types (faq, userGuide): `{domain}/rcs/v1/siteIds/citi-china/akamaiCDN/contentTypes/{contentType}.json`
Plus any URLs defined in SiteSetting > API > `akamaiPurgeUrl` with `{domain}`, `{siteId}`, `{type}`, `{businessId}` placeholders.

Manual Cloudflare action: After publishing any CDN-enabled content for `citi-china`, purge all the above URL variants from Cloudflare. If `akamaiAdditionalDomains` is configured in SiteSetting > API, purge across all domains.

Source: `AkamaiFeedContentQueryService.java:78-86,120-134,156-165`

4. Withdraw Content (CDN-Enabled Content Types) IF enableQueryFromCDN

CMS Steps:

1. User opens a published `citi-china` content item
2. User clicks **Withdraw**
3. CMS sends a Kafka message (`EVENT_WITHDRAW`, resource type `content`)
4. `AkamaiFeedContentQueryService` purges the CDN URLs BROKEN

What breaks: Withdrawn content remains accessible via CDN JSON API URLs, exposing data that should no longer be public for the Citi China site.

Manual Cloudflare action: Same as Section 3 — purge all CDN URL variants for the withdrawn content item.

Source: `AkamaiFeedContentQueryService.java:40-42`

5. Publish an Image/Asset (AKNS-Enabled) IF enableAKNS

CMS Steps:

1. User uploads or publishes an image in the `citi-china` Assets Manager
2. CMS sends a Kafka message (`EVENT_PUBLISH`, resource type `media`)
3. `AkamaiNSFeedService` checks `enableAKNS=true` in SiteSetting > Assets for `citi-china`
4. If enabled, it uploads the image to Akamai NetStorage at `/v1/aknetpublic/citi-china/assets/{fileId}.jpg` BROKEN
5. The image is normally served via `/rcs/citi-china/akpublic/storage/{businessGroup}/{businessId}`

What breaks: The image will not be uploaded to CDN origin storage. The `akpublic/storage/` URL will return 404 or stale content. Share links using the `enableAkamai=true` path will not work for new images. A failure notification email is sent to the asset uploader or the AKNS notification BCC list.

Manual Cloudflare action:

- Ensure the image is accessible via the non-CDN path (`/rcs/citi-china/storage/...`) as a fallback.
- If CDN-served paths are required, manually upload to Cloudflare R2 storage (or equivalent).
- If the image was previously cached, purge the asset URL from Cloudflare.

Source: `AkamaiNSFeedService.java:91-106,118-175`

6. Withdraw an Image/Asset (AKNS-Enabled) IF enableAKNS

CMS Steps:

1. User withdraws an image from the `citi-china` Assets Manager
2. CMS sends a Kafka message (`EVENT_WITHDRAW`, resource type `media`)
3. `AkamaiNSFeedService` attempts to delete the file from AKNS and purge CDN cache BROKEN

What breaks: The withdrawn image remains on CDN storage and continues to be publicly accessible via the Citi China domain.

Manual Cloudflare action:

- Purge the asset URL from Cloudflare: `https://{citi-china-domain}/EPFPublicService/v1/aknetpublic/citi-china/assets/{fileId}.jpg`
- Manually delete the file from Cloudflare storage (R2) if applicable.

Source: `AkamaiNSFeedService.java:176-246`

7. Regenerate Static Pages (SSG Admin) IF SSG ENABLED

CMS Steps:

1. Admin navigates to **SSG Admin > Static Page Regeneration** tab
2. Admin selects `citi-china` pages and/or content types
3. Admin clicks **Regenerate**
4. SSG re-generates static HTML for selected pages
5. `AkamaiFastPurgeService` invalidates the regenerated page URLs **BROKEN**

What breaks: Regenerated pages will have updated HTML on the origin server, but Cloudflare CDN continues to serve the old cached version.

Manual Cloudflare action: After regeneration completes, purge all regenerated page URLs from Cloudflare. For bulk regeneration of `citi-china`, consider using **"Purge Everything"** for the Citi China zone.

Source: `SsgAdmin.js:321-355`, `AkamaiFastPurgeService.java:130-133`

8. Change Page URL (Clear Old URL) IF SSG ENABLED

CMS Steps:

1. User changes the `pageUrl` of a `citi-china` page in Page Builder
2. User publishes the page with the new URL
3. CMS sends a Kafka message with `EVENT_CLEAR_OLD_URL`
4. SSG deletes HTML at the old URL and generates HTML at the new URL
5. Akamai purges both old (delete) and new (invalidate) URLs **BROKEN**

What breaks: The old URL continues to serve cached content on the Citi China Cloudflare zone. The new URL may serve stale or no content.

Manual Cloudflare action: Purge **both** the old page URL and the new page URL from Cloudflare cache.

Source: `SsgPageKafkaConsumer.java:184-185`

9. TTS PDF/Asset Feed Publication IF TTS CONTENT TYPES EXIST

CMS Steps:

1. TTS content (faq, userGuide) for `citi-china` is published, triggering PDF generation
2. `TtsPdfServiceImpl` generates a PDF and calls `AssetAkamaiServiceImpl.purgeAkamaiCacheForBusinessIdLink()` to purge the asset share link **BROKEN**
3. The purge uses the `externalDomain` from `citi-china`'s global SiteSetting and also checks `akamaiPurgeStagingDomains` / `akamaiPurgePublicDomains`

What breaks: Updated PDFs and support assets remain stale on CDN. The share link URLs are not purged.

Manual Cloudflare action: Purge the specific asset share link URLs:

`/rcs/citi-china/storage/{businessGroup}/{path}/{businessId}`
Also purge the TTS content query URLs if applicable (see Section 3 patterns).

Source: `TtsPdfServiceImpl.java:96`, `AssetAkamaiServiceImpl.java:29-97`

10. Admin Manual Cache Purge (Akamai Admin Page) APPLIES

CMS Steps:

1. Admin navigates to **Akamai Admin** page in CMS UI
2. Admin enters a `citi-china` URL and clicks **Refresh**
3. CMS calls `/akamaiFastPurge/icgPublic/deleteCache` or `/siteIds/citi-china/akamaiNS/upload` BROKEN

What breaks: The entire manual purge feature in CMS admin is non-functional for Citi China. Admins cannot purge Cloudflare cache through the CMS interface.

Manual Cloudflare action: Use the Cloudflare dashboard directly. Navigate to **Caching > Purge Cache > Custom Purge**, and enter the URLs for the Citi China zone.

Source: `AkamaiAdmin.js:25-59`

11. Admin Upload to Akamai NetStorage APPLIES

CMS Steps:

1. Admin navigates to **Upload Akamai** page in CMS UI
2. Admin selects a CP Name (`RetailPrivatePagesService`, `CitigroupUI`, or `speedUi`)
3. Admin uploads a ZIP file
4. CMS uploads the ZIP contents to Akamai NetStorage BROKEN

What breaks: Static assets (JS, CSS, HTML packages) for Citi China cannot be uploaded via CMS to CDN storage.

Manual Cloudflare action: Manually upload the files to Cloudflare R2 storage (or equivalent origin) via the Cloudflare dashboard or `wrangler` CLI tool.

Source: `UploadAkamai.js:12-97`

12. citicms-repo Deployment (upload.sh) APPLIES

CMS Steps:

1. During deployment, `upload.sh` zips the `aknetpublic/` directory
2. Script uploads the ZIP to Akamai NetStorage via `/siteIds/rpt/akamaiNS/upload/fileTypes/zip?cpName=RetailPrivatePagesService` BROKEN

What breaks: Deployment of repo component assets (JS/CSS bundles used by `citi-china`, served from `/chn/chn-ui/akpublic`) to CDN storage fails silently or errors out.

Manual Cloudflare action: After each deployment, manually upload the `aknetpublic/` directory contents to Cloudflare storage and purge the corresponding URLs under the `/chn/chn-ui/akpublic` path.

Source: `upload.sh:24-39`, `startCitigroupUi.sh:62`

13. Akamai/MongoDB Data Sync IF enableAKNS

CMS Steps:

1. Admin triggers a data comparison between MongoDB and Akamai NetStorage for `citi-china`

2. `AkamaiDataCompareServiceImpl` compares asset counts and IDs between Mongo and AKNS **BROKEN**
3. `AkamaiDataSyncServiceImpl` uploads missing assets to AKNS or deletes redundant ones, then purges CDN cache **BROKEN**

> **What breaks:** The data sync tool that ensures MongoDB and CDN storage are in sync is completely non-functional for the Citi China Cloudflare zone.

> **Manual Cloudflare action:** Manually verify that all published `citi-china` assets in MongoDB also exist in Cloudflare storage. No automated reconciliation is available.

Source: `AkamaiDataSyncServiceImpl.java:30-69`, `AkamaiDataCompareServiceImpl.java:31-126`

14. Bulk Data Update (Visual Assets Update) APPLIES

> **New finding:** This workflow was not in the original document. It was identified during code validation against the `citi cms-service` codebase.

CMS Steps:

1. Admin performs a bulk/visual data update for `citi-china` content (e.g., find-and-replace across assets)
2. `VisualAssetsDataUpdateService` processes the update requests
3. If asset redirect mappings are created, the service calls `akamaiFastPurgeClient.deleteCache()` to purge the affected page URLs **BROKEN**

> **What breaks:** After a bulk data update that creates redirect mappings for `citi-china` assets, the old URLs remain cached on Cloudflare, causing redirects to not take effect.

> **Manual Cloudflare action:** After performing a bulk data update for `citi-china`, purge all affected page URLs from Cloudflare. The URLs are listed in the `DataUpdateRequest.pageUrl` field of each update request.

Source: `VisualAssetsDataUpdateService.java:232`

Summary: Quick Reference Table for Citi China

#	CMS Action	Condition	What Fails	Manual Cloudflare Step
1	Publish Page	SSG enabled	Stale page served up to 7 days	Purge page URL
2	Withdraw Page	SSG enabled	Withdrawn page still accessible	Purge page URL
3	Publish CDN Content	enableQueryFromCDN	Stale JSON on CDN API endpoints	Purge all content API URL variants
4	Withdraw CDN Content	enableQueryFromCDN	Withdrawn content still on CDN	Purge all content API URL variants
5	Publish Image/Asset	enableAKNS	Image not uploaded to CDN storage	Upload to Cloudflare R2 + purge
6	Withdraw Image/Asset	enableAKNS	Deleted image still on CDN	Purge asset URL + delete from R2
7	Regenerate Pages	SSG enabled	Regenerated pages cached stale	Purge all regenerated URLs
8	Change Page URL	SSG enabled	Old URL cached, new URL stale	Purge both old and new URLs
9	TTS PDF/Asset Feed	TTS types exist	Stale PDFs/assets on CDN	Purge specific asset URLs
10	Admin Manual Purge	Always	Feature non-functional	Use Cloudflare dashboard directly
11	Admin Upload to AKNS	Always	Upload feature non-functional	Upload to Cloudflare R2 manually
12	Repo Deployment	Always	Asset deployment to CDN fails	Upload to Cloudflare R2 manually
13	Data Sync Check	enableAKNS	Sync tool non-functional	Manual verification needed
14	Bulk Data Update	Always	Redirect mappings cached stale	Purge affected page URLs

Citi China Domains to Monitor

The CDN cache purge is domain-aware. For `citi-china`, the domains that need manual purging are configured in:

- `SiteSetting > Environment > domain` — the primary public domain for Citi China
- `SiteSetting > Environment > previewDomain` — the preview/staging domain
- `SiteSetting > API > akamaiAdditionalDomains` — any additional domains configured
- `SiteSetting > API > akamaiPurgeStagingDomains` — staging-specific purge domains
- `SiteSetting > API > akamaiPurgePublicDomains` — public-specific purge domains
- `Global SiteSetting > externalDomain` — used by `AssetAkamaiServiceImpl` for share link purges

Asset prefix path: `/chn/chn-ui/akpublic`

Source: `SSGConfigurationChecker.java:152-158`, `AssetAkamaiServiceImpl.java:49-94`, `startCitigroupUi.sh:62`

Workflows Validated as Not Applicable to Citi China

The following Akamai-dependent workflows were found in the codebase but are **not applicable** to the `citi-china` siteId:

Workflow	Why Not Applicable	Source
Sitemap Generation	Hardcoded to `citigroup.com` and `icg.citi.com` domains only	`SitemapGenerationServiceImpl.java:441-448`

Purge		
GPA/GPS Insights Feed Purge	Applies only to `citigpa` / `icgPublic` siteIds	`GPAInsightsServiceImpl.java`, `ICGInsightsServiceImpl.java`
GPA Staging Insights Feed Purge	Applies only to `citigpa` staging siteId	`GPAStagingInsightsServiceImpl.java`