

The Site Migration and Redirect-Mapping Playbook

How to inventory URLs, build a redirect map, test it on staging, run launch day, and monitor the move in Search Console when a domain, CMS, or URL structure changes.

Contents

01 Work out which kind of move you are making

02 Build the URL inventory from more than one source

03 Write the redirect map so every row has a reason

04 Prepare the new site and test it before anyone links to it

05 Launch and monitor: the step-by-step checklist

06 After launch: what to watch, and what normal looks like

Key Takeaways & Sources

01

Work out which kind of move you are making

Google splits site moves into two documents, and the split matters for planning. A move with URL changes covers HTTP to HTTPS, a new domain, merged hostnames, and any rewrite of paths. A move without URL changes covers infrastructure work such as new hosting, where every address stays the same. The first needs a redirect map and, for domain changes, a Change of Address submission. The second mostly needs careful DNS and server testing. Treating a URL-changing project as an infrastructure project is how redirect work gets skipped.

The next decision is scope. Google recommends changing one thing at a time: if you want a new domain, a new CMS, and a new layout, do them in sequence rather than together. A combined launch is common because it saves project time, but it makes diagnosis hard. If organic traffic falls afterwards, you cannot say whether the redirects, the template change, or the content edits did it. If the business cannot wait, at minimum keep the URL structure and on-page content stable during a CMS or design change, so the only variable is the platform.

Third, decide whether to move all at once or in sections. Google's guidance is that small and medium sites should move all URLs together, which helps its systems detect the move faster, and that large sites can move in sections to make problems easier to find and fix. If you split a large site, choose a first section that changes rarely and is not exposed to seasonal swings. Be aware that one section is a useful test but not a full preview of a whole-site move: the more pages you move, the more problems appear.

- URL changes (domain, HTTPS, paths, merged hostnames): redirect map, canonical updates, and Change of Address for domain or subdomain moves
- No URL changes (hosting, infrastructure): server and DNS testing, with robots.txt and verification carried over
- Mixed projects: split into stages, or freeze URLs and content while the platform changes

Google recommends changing one thing at a time: if you want a new domain, a new CMS, and a new layout, do them in sequence rather than together.

02

Build the URL inventory from more than one source

A crawl of the live site is the usual starting point, but Google's documentation lists several other places to find old URLs, and each catches something the crawl does not. Sitemaps show what you have told Google matters. Server logs and analytics show which URLs get visited, including orphaned pages that no internal link reaches. The Links report in Search Console shows pages that attract external links, which are the ones whose loss hurts most. Your CMS can export every URL that hosts content. Merge the sources and de-duplicate; the union is your inventory.

Include non-HTML files. Google specifically says to cover images, videos, JavaScript, CSS, and downloads such as PDFs, because they may already receive search traffic or links and need to be moved the same way as pages. This is the usual gap in a page-only inventory: a white paper PDF with a decade of backlinks gets left out of the map and returns a 404 on launch day.

Once the inventory exists, add three columns that make the map decidable: a traffic or impressions figure from a period you choose with seasonality in mind, an inbound link count, and a business owner. Rank by the first two. You will not hand-review tens of thousands of low-value URLs, but you should be able to defend every decision about the top few hundred, and you should know who signed off on pages being merged or dropped.

This is the usual gap in a page-only inventory: a white paper PDF with a decade of backlinks gets left out of the map and returns a 404 on launch day.

03

Write the redirect map so every row has a reason

Each old URL gets one of four outcomes: a one-to-one redirect to its direct equivalent on the new site, a redirect to a consolidated page where several old pages were merged, a deliberate 404 or 410 because the content is gone, or no change because the URL stays the same. Google says content that is not carried over should return a proper 404 or 410 on the new site, and that redirecting many old URLs to one irrelevant destination, with the home page the example it gives, can confuse users and may be treated as a soft 404. Redirecting several old pages to one new page is fine when that page genuinely consolidates them.

Use pattern rules where the structure allows it and explicit rows where it does not. A domain change with identical paths can be a single wildcard rule. A CMS change that rewrites every slug needs a lookup table. Watch for chains: Google says Googlebot can follow up to ten hops, but advises redirecting straight to the final destination, and if that is impossible, keeping chains low, ideally no more than three hops. The usual source of chains in a migration is a legacy redirect that already existed. Resolve those by pointing the original old URL at the final new URL.

Use permanent server-side redirects, 301 or 308, as Google recommends. Client-side redirects are a fallback when server-side ones are impossible, not a design choice. Normalise the variants as well: HTTP and HTTPS, www and non-www, trailing slash and none, uppercase paths. Each one should reach the new canonical URL in a single hop. Finally, decide the fate of query-string URLs, including parameters used for tracking and filters. They are the rows most often forgotten and the most likely to create duplicate paths on the new site.

Old URL situation	Recommended handling	Check before launch
Page exists on the new site at a new address	301 or 308 to the exact equivalent	Destination returns 200 and is self-canonical
Several old pages merged into one	Redirect each to the consolidated page	The new page really covers the merged intent
Content retired with no replacement	Return 404 or 410 on the new site	No internal links or sitemap entries still point at it
Image, PDF, or other file	Redirect to the file's new address	File opens and was not blocked by robots.txt
Parameterised or tracking URL	Redirect to the clean URL, or let it resolve to the page	No second indexable version is created
Existing redirect from an earlier project	Repoint the original URL at the final destination	No chain longer than one hop

04

Prepare the new site and test it before anyone links to it

On the new site, three on-page items change for every moved URL. Each page needs a self-referencing `rel=canonical` pointing at its new address. If the site uses `hreflang`, the annotations must be updated to the new URLs. Internal links should be rewritten to the new addresses, using the redirect map to find them, so that users and crawlers do not take a redirect on every click inside your own site. The new sitemap should contain only the new URLs from the map.

Staging environments are where migrations most often go wrong, in a predictable way. Teams block crawling or add `noindex` while the site is in development, which is sensible, and then ship the block to production. Google lists this first among common mistakes. Before launch, write down what `robots.txt` should say on day one, and list every template where a `noindex` directive has to be removed. Confirm too that a missing `robots.txt` returns a real 404 rather than a generic page. Also check that Search Console verification survives the move: the HTML file method, meta tag, and analytics-based methods all depend on the new platform carrying the verification token across.

Test the map, not a sample of it. Run a crawler over the full list of old URLs against staging or a hosts-file-pointed copy of production, and record the status code, final destination, hop count, and

whether the destination is indexable. Fail anything that returns something other than a single redirect to a 200 page, unless the row is a planned 404 or 410. Google also notes that Googlebot will temporarily crawl a migrated site more heavily than usual, since it fetches both old and new URLs, so confirm the new hosting can take the extra load and warn the host if the site is large.

Staging environments are where migrations most often go wrong, in a predictable way.

05

Launch and monitor: the step-by-step checklist

Schedule the launch for a low-traffic window if your traffic has a pattern, because fewer people meet any fault and more server capacity is available for crawling. The sequence below treats each step as a gate. If a gate fails, the answer is to fix it before the next step rather than to carry on and note it for later.

Step	Owner	Gate to pass before moving on
Verify every old and new property variant in Search Console, including www and non-www, HTTP and HTTPS	SEO lead	Each variant shows as verified and survives the move of the verification token
Freeze the redirect map and the list of top URLs by traffic and links	SEO lead with content owner	Map is signed off; every top URL has a named destination or a documented retirement
Crawl the full old-URL list against the staging build	SEO or QA engineer	Every row returns one redirect hop to the expected 200 page, or the planned 404 or 410
Remove staging blocks: noindex tags, robots.txt disallows, password gates	Developer	URL Inspection on sample new URLs shows they are indexable and crawlable
Switch redirects live and deploy the new site	Developer	Spot-check of the top 50 URLs on production returns the expected status and destination
Submit Change of Address for the old domain-level property (and separately for any subdomain that moved) if the domain or subdomain changed	SEO lead	Submission accepted; the tool covers all protocols of the property but not its subdomains, and is not needed for HTTPS, www switches, or path-only changes
Submit the new sitemap, keep the old sitemap for monitoring	SEO lead	Search Console accepts both and shows new URLs discovered

Step	Owner	Gate to pass before moving on
Update internal links, profile links, ad destinations, and high-value external links	Marketing and PR	Paid campaigns and social profiles point at new URLs; top linking sites contacted
Review server logs, error rates, and analytics in the first hours and days	Developer and analyst	No unexpected 4xx or 5xx spikes; Googlebot is fetching new URLs

06

After launch: what to watch, and what normal looks like

Google's own description of a healthy migration is gradual: indexed URLs on the old property fall, indexed URLs on the new one rise, and the sitemap reports reflect that over time. It says the process runs per URL, that a small to medium site can take a few weeks for most pages to move, that larger sites take longer, and that rankings may fluctuate temporarily before settling. Search Console may also warn about redirecting URLs in the old sitemap; Google says to ignore that, since you are moving the URLs deliberately. Plan reporting around those expectations so a dip in week one does not trigger a panic rollback.

The checks worth running weekly for the first quarter are narrow. Compare Search Console performance for old and new properties by page group. Look for a rising count of not-found or redirect-related errors in the Pages report. Re-crawl the top URLs to confirm redirects still work, since deploys can break them. Inspect a handful of URLs that look missing from Google, using URL Inspection to see what Googlebot actually received. Where a section is slow to move, the cause is nearly always something concrete, such as a canonical pointing at the old URL, a stray noindex, or a redirect to a URL that does not exist.

Keep the redirects. Google recommends at least a year, because that is how long it can take to recrawl and reassign signals from links on other sites, and suggests considering indefinitely for users. Tidy the highest-volume external links so they point directly at new URLs, since redirects are slower for people. This playbook cannot promise a particular recovery curve or tell you how much traffic a given migration will cost. Those depend on size, server speed, and how much else changed, and any specific percentage offered in advance should be treated with suspicion.

- Week one: daily checks of errors, logs, and top-URL redirects
- Weeks two to six: weekly comparison of old and new property performance by page group
- Months two to twelve: monthly redirect re-crawl and a review of any section still indexed under old URLs
- After a year: decide whether to retire low-traffic redirects, and keep any that still receive visits or links

Key Takeaways

- Change one thing at a time. Google advises against moving domain, CMS, and layout in the same release, because you lose the ability to tell which change caused a problem.
- The redirect map is the project. Every old URL that earns traffic or links needs a deliberate, relevant destination, decided before launch rather than during it.
- Redirect to the closest equivalent page, not to the home page. Mass redirects to one irrelevant URL can be treated as soft 404s.
- Test on staging with a crawler against the full map, then re-test on production within the first hour. Most migration damage is a redirect or indexing block nobody checked.
- Keep redirects for at least a year, and expect ranking fluctuation while Google reprocesses URLs. A dip in week one is not, by itself, a failed migration.

Sources

1. Site moves with URL changes (Google Search Central) —
<https://developers.google.com/search/docs/crawling-indexing/site-move-with-url-changes>
2. Site moves without URL changes (Google Search Central) —
<https://developers.google.com/search/docs/crawling-indexing/site-move-no-url-changes>
3. Redirects and Google Search (Google Search Central) —
<https://developers.google.com/search/docs/crawling-indexing/301-redirects>
4. Introduction to robots.txt (Google Search Central) —
<https://developers.google.com/search/docs/crawling-indexing/robots/intro>
5. Change of Address tool (Search Console Help) — <https://support.google.com/webmasters/answer/9370220>