

STANDARD OPERATING PROCEDURE · WW-008

Plan, implement and test 301 redirects for changed URLs

Map every changed or removed URL to its best replacement, implement permanent redirects without chains, and verify that rankings and links carry over.

TECHNICAL SEO

VERSION 1.0	EFFECTIVE 5 October 2026
NEXT REVIEW 5 April 2027	TIME 2 hours
DIFFICULTY Advanced	RELATED WW-004
AUDIENCE Developers and SEO specialists handling URL changes, restructures or rebuilds	
TOOLS Google Search Console · Google Sheets · Website CMS, server or hosting redirect manager · Terminal or online redirect checker	

Purpose

When a URL changes without a redirect, its rankings, backlinks and bookmarks are lost. This procedure builds a complete one-to-one redirect map, implements it as permanent (301) redirects, and tests that every old URL lands on the right page in a single hop. It is finished when every mapped URL passes testing and Search Console shows no new 404 errors for old URLs after two weeks.

Scope

In scope

- Inventorying old URLs from the sitemap, Search Console and analytics
- Mapping old URLs to the most relevant new URLs
- Implementing 301 redirects and updating internal links and sitemaps
- Testing before and after go-live, and monitoring

Out of scope

- Domain changes requiring the Search Console Change of Address tool (follow Google's site move guidance alongside this SOP)
- Server configuration beyond adding redirect rules
- Content changes on the new pages

Roles and responsibilities

Role	Responsibility
SEO specialist	Builds and signs off the redirect map; monitors Search Console.
Developer	Implements redirects, updates internal links and the sitemap.
Client	Confirms which retired pages have no replacement.

Prerequisites

- A list of the URL changes planned (restructure, rebuild or individual page moves).
- Search Console access (WW-004) and analytics access.
- Access to the redirect mechanism: CMS redirect manager or plugin, hosting control panel, or server configuration.
- A redirect map sheet with columns `Old URL` , `New URL` , `Status` , `Tested` , `Notes` .

Procedure

Inventory every old URL

A redirect map is only as good as its list of old URLs.

1. Export URLs from the current XML sitemap into the `Old URL` column.
2. In Search Console, export the **Pages** list from **Performance** for the last 12 months, and add any URLs not already listed.
3. In analytics, export landing pages for the last 12 months and add missing URLs. Remove duplicates.

[!IMPORTANT] Capture this inventory before the old site is changed. Once old pages are gone, the list is much harder to rebuild.

Map old to new

Send each old URL to the single most relevant replacement.

1. For each old URL, enter the closest equivalent new URL. Match like for like: a service to the same service, a post to the same post.
2. Where content is retired with no equivalent, map it to the closest parent page (for example the category or services page) and note why. Only leave a URL to return 410 or 404 if nothing relevant exists.
3. Make sure no new URL is itself an old URL in the map, which would create a chain.

[!WARNING] Do not redirect everything to the home page. Google treats irrelevant redirects as soft 404s and the ranking value is lost.

Implement the redirects

Add permanent redirects in one place.

1. Add each mapping as a **301 (permanent)** redirect using a single mechanism: the CMS redirect manager, the hosting redirect tool, or server rules. Avoid splitting rules across several layers.
2. Preserve query strings where the platform supports it, and match trailing slash and case consistently.
3. Update internal links, navigation menus and canonical tags so they point straight to the new URLs rather than relying on redirects.
4. Regenerate the XML sitemap so it lists only new URLs.

Test every redirect

Prove each old URL reaches its destination in one hop.

1. Test each old URL using a redirect checker or the terminal command `curl -sI <old-url>`, and confirm the response is `301` with the `location` header set to the mapped new URL.
2. Confirm the new URL returns `200` and that no URL passes through more than one redirect.
3. Mark each row `Pass` or `Fail` in the **Tested** column and fix every failure before go-live is signed off.

Monitor after go-live

Catch anything the map missed.

1. Submit the new sitemap in Search Console (WW-004).
2. Check **Indexing**, then **Pages**, for the **Not found (404)** reason weekly for four weeks, and add redirects for any old URLs that appear.
3. Keep the redirects in place for at least one year, and preferably permanently.

Quality checks

- ☐ Every URL in the inventory has a mapped destination or a documented reason for none.
- ☐ Every redirect returns `301` and lands on a `200` page in a single hop.
- ☐ Internal links, menus and canonicals point to new URLs directly.
- ☐ The sitemap lists only new URLs and has been resubmitted.
- ☐ No new 404s for old URLs appear in Search Console after two weeks.

Troubleshooting

Symptom	Likely cause	Fix
Redirect returns <code>302</code>	Temporary redirect type selected	Change the rule to <code>301</code> permanent.
Two or more hops	Old redirects point to URLs that now redirect again	Update the earlier rules to point straight to the final URL.
Redirect loop	Source and destination match after normalisation (slash or case)	Make trailing slash and case handling consistent.
Rankings drop for weeks	Redirects to irrelevant pages, or content removed	Re-map to closer equivalents and restore missing content.

Records

- Keep the signed-off redirect map with test results in the client's project folder.
- Record the go-live date and the four weekly 404 checks.

References

- [Redirects and Google Search \(Google Search Central\)](#)
- [Site moves with URL changes \(Google Search Central\)](#)
- [Page indexing report \(Search Console Help\)](#)

