

# OverviewSitemapStructure

## Lenya Sitemap Structure

Lenya is a Cocoon application. The central Cocoon configuration file is the sitemap. In a nutshell, the sitemap defines a set of rules that describe what output will be sent back to the user's browser in response to a given request. For more details on the Cocoon sitemap, refer to the [Cocoon documentation](#), especially the parts about the [sitemap configuration](#).

### Nested sitemaps

In a quite simple Cocoon application there is just one sitemap.xmap file. But Lenya is not necessarily what you would consider a "simple" Application. Therefore the sitemap in Lenya is spread over a couple of files inside the lenya webapp directory.

There is a number of reasons why there is not just one sitemap in Lenya but actually a tree of sitemaps. You might guess of of those reasons being easier debugging and a better learning curve. But the opposite is the case. The fact that a request is passed down a tree of sitemaps does not make it easier to understand in the first place, as you can guess by reading [GetPageWalkthrough](#).

It might seem to be easier to put everything just in one sitemap, but there are good reasons to have nested sitemaps despite the extra complexity:

- Separation of Lenya CMS and publication content
- Separation of Lenya core and publication specific parts
- Implementation of fallback mechanisms (one of the major design principles in Lenya)
- Diverting to sub-sitemaps based on values of sitemap variables at runtime for a particular request (example: document types)

### Understanding <map:mount ...> tags

The mechanism for mounting sub-sitemaps is the <map:mount ...> tag in sitemap.xmap. See: [Cocoon Documentation -%3E Concepts -> Sitemap](#) (search for the "Mounting sitemaps" headline).

\*Note: Keep in mind that a sub-sitemap must either end the request (by calling some serializer) or pass it to a sitemap which is one level deeper. But there is no mechanism to pass the request back up the chain to a higher level. If the request is passed to a sub-sitemap which does not contain a matching pipeline which ends in a serializer, an exception will be thrown that no pipeline matches the request.

The only mechanism available to you in a sub-sitemap to make a call to a pipeline at a higher level is to use the cocoon: subprotocol, as each sub-request starting with cocoon:// will be started to be processed in the root sitemap.xmap as if it was an entirely new request.\*

### The actual sitemap tree in Lenya

*This section is not up-to-date wrt Lenya 2.0. – JörnNettingsmeier <<DateTime(2007-07-30T13:46:54Z)>>*

- sitemap.xmap (the main sitemap)
    - lenya/i18n.xmap (internationalization)
    - global-sitemap.xmap
      - fallback:lucene.xmap
      - lenya/lenya.xmap
      - lenya/pubs/<yourpub>/menus.xmap
      - lenya/navigation.xmap
      - lenya/pubs/<yourpub>/parameter-\*.xmap
      - lenya/pubs/<yourpub>/policies-sitemap.xmap
      - lenya/admin.xmap (admin area)
      - lenya/scheduler.xmap
      - lenya/info.xmap (info area)
      - lenya/pubs/<yourpub>/sitemap.xmap
        - lenya/pubs/<yourpub>/publication-sitemap.xmap
        - lenya/pubs/<yourpub>/doctypes.xmap
      - lenya/usecase.xmap (use cases)
        - usecase-\*.xmap
      - lenya/resources.xmap
      - lenya/resources-shared.xmap
- Note: there are two files named "sitemap.xmap": one at the Lenya level, and another within your publication directory.  
Note: This page has a detailed analysis of sitemap processing attached to it:
- [sequence.html](#)