

# Redirection (2.1 legacy document)

## Table of contents

1 Comments.....	4
-----------------	---

## Table of contents

1 Introduction.....	3
2 HTTP redirects and how they work.....	3
3 Internal Redirects Using the Cocoon Protocol.....	3
4 Session Management with Redirects.....	3
5 Temporary and Permanent Redirects.....	3
6 Redirects in Pipelines.....	4
7 Global Redirects.....	4
8 Redirecting to Resources.....	4

**Warning:**

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

## 1. Introduction

A Redirector allows the sitemap to pass a request for one URI on to another, whether that other URI is handled by Cocoon or not.

To redirect from page1.html to page2.html, you can use the following:

```
<map:match pattern="page1.html"> <map:redirect-to uri="page2.html"/> </map:match>
```

## 2. HTTP redirects and how they work

If the URI specified does not use the Cocoon: protocol, then an HTTP redirect will occur. The new URI is sent back to the client, which will then request the page from this new location.

Therefore, directory handling in redirect URIs works differently from other sitemap components.

If the new URI is relative, then it will be relative to the directory of the page that called it, not relative to the URI of the sitemap containing it. Thus, the following is incorrect:

```
<map:match pattern="folder/page1.html"> <map:redirect-to uri="folder/page2.html"/>
</map:match>
```

This will in fact redirect the user to folder/folder/page2.html, which is probably not intended. The correct version is:

```
<map:match pattern="folder/page1.html"> <map:redirect-to uri="page2.html"/> </map:match>
```

## 3. Internal Redirects Using the Cocoon Protocol

A redirection URI can make use of the cocoon: protocol to return content from another Cocoon pipeline. In this case, the redirection happens internally. The content from the redirected URI is returned to the client as if it came from the original URI.

Directory handling is the same here as for other sitemap components. So that:

```
<map:match pattern="folder/page1.html"> <map:redirect-to uri="cocoon:/folder/page2.html"/>
</map:match>
```

will return the content of page2.html to the client in response to the request for page1.html.

Note: when the cocoon: protocol is used, an HTTP redirect is not used.

## 4. Session Management with Redirects

By setting the session attribute to yes, the current session will be maintained during the redirect.

## 5. Temporary and Permanent Redirects

By default, an HTTP redirect sends a code of SC\_MOVED\_TEMPORARILY, (302). This instructs the user agent to use the new URI, but not to cache the resulting page, as it may well soon revert back to the old URI.

This can be a problem for pages that have been moved permanently, as the new page will never be cached, placing additional load on both the browser and on the server.

This can be avoided using a permanent redirect, using a code of SC\_MOVED\_PERMANENTLY (301). A permanent redirect can be specified as:

```
<map:match pattern="page1.html"> <map:redirect-to uri="page2.html" permanent="yes"/>
</map:match>
```

This results in the user agent caching the redirected page, and thus saves resources both on the server and for the client's browser.

## 6. Redirects in Pipelines

A redirect must stand alone in a pipeline - it cannot occur after a generator. If a redirect needs to be generated conditionally by a pipeline, then a <meta> tag redirect should be added into the <head> of the HTML page. The syntax for this is:

```
<html> <head> <meta http-equiv="refresh" content="0;URL=page2.html"/> ... </head> ...
</html>
```

## 7. Global Redirects

When an aggregator accesses a source that includes a redirection, it will aggregate the document specified by the redirection URI.

Alternatively, if a redirection that has the global attribute is set (to yes or true) occurs within an aggregation, the aggregation is cancelled and the redirect is sent back to the client.

## 8. Redirecting to Resources

Specifying a resource attribute allows the redirection to a sitemap resource. This usage has been deprecated. map:call should be used instead.

## 1. Comments

add your comments