Reader

A mini-pipeline – this Components functions as a complete self-contained pipeline.

The simplest implementation is the Resource Reader, which simply reads a file from disk and streams it back to the user#1.

Declaration

```
<map:readers default="resource">
  <map:reader name="resource" src="org.apache.cocoon.reading.ResourceReader"/>
</map:readers>
```

Usage

```
<map:match pattern="sites/images/*.gif">
  <map:read src="resources/images/{1}.gif" mime-type="image/gif"/>
</map:match>
```

Note that the `map:read` element requires a `mime-type` attribute that indicates the mime-type of the file being served#2.

See also ServingStaticFiles

Available Implementations

- ResourceReader – simply reads files from disk and streams them back in the HTTP response.
- JSPReader – Takes information from a JSP page?
- DatabaseReader – reads a resource from a database (i.e. a BLOB). Is configured with the connection parameter, table and column name
- ImageReader – Is able to resize image files on-the-fly

Readers Comments

Note that ResourceReader actually uses Sources. So it's by far not restricted to files. The SourceResolver will automatically choose which Source to use in order to aquire the data by looking at the contents of the src attribute. AFAIK there are Sources for reading from the file system, from the CLASSPATH, from ftp and from http plus it's easy to add your own.

Note that at least for Cocoon 2.1 the mime-type can also be determined by the Source, if it is able to provide it. The mime-type attribute can be used to override this. Otherwise the mapping from extensions to mime-types of your servlet container (Tomcat: web.xml) will be used.