

Restore

Table of contents

1 Introduction.....	2
2 Implementation.....	2
3 Parameters.....	2
4 usecase.....	2
4.1 Delete screen.....	2
4.2 Restore confirmation.....	2
5 Ant Task	3

1. Introduction

The restore operation corresponds to a move operation from an area, trash or archive, to the authoring

2. Implementation

Similar to the operation [move](#) (move.html)

3. Parameters

Required parameters:

- the area for the source document
- the document id for the source document
- the task id

4. usecase

4.1. Delete screen

URL :

```
{document-URL}?lenya.usecase=restore&lenya.step=showscreen
```

usecase sitemap:

```
<map:match pattern="restore" type="usecase">
  <map:match pattern="showscreen" type="step">
    <map:generate src="content/info/restore.xsp" type="serverpages"/>
    <map:transform src="xslt/info/restore.xsl">
      <map:parameter name="use-request-parameters" value="true"/>
    </map:transform>
    <map:call resource="style-cms-page"/>
  </map:match>
</map:match>
```

The parameters for the source are get with the serverpage through the page envelope input module. The destination document id is computed in the serverpage from the identifier. The destination source is set to area in the stylesheet. A form (build with the xslt transformation) sends then the parameters as request parameters with the new URL.

4.2. Restore confirmation

URL :

```
{document-URL}?lenya.usecase=restore&lenya.step=restore&...{source parameters}
```

usecase sitemap:

```

<map:match pattern="restore" type="usecase">
  <map:match pattern="restore" type="step">
    <map:act type="task">
      <map:redirect-to session="true" uri="{request:contextPath}/{../../../../1}/{../../../../2}"/>
    </map:act>
  </map:match>
</map:match>

```

The action `org.apache.lenya.cms.cocoon.acting.TaskAction` calls the execution of the ant task.

5. Ant Task

The ant target `restoreDocument` is in the publication :

```
{publication}/config/tasks/targets.xml
```

The target for the restore operation depends on the different targets

- `firstareaproperties`, to set the needed properties dependent of the area of the source
- `secareaproperties`, to set the needed properties dependent of the area of the destination
- `newcopydocumentid`, to compute the unique destination id (Same logic like for the copy operation).
- `firstdocumentpath`, to compute the directory of the contents for the source (Needed for the revisions and the rcml files)
- `secdocumentpath`, to compute the directory where are the contents for the destination (Needed for the revisions and the rcml files)
- `move`, to execute the different move operations

More about ant task, see the documentation [Ant Task](#) (`../tasks/anttask.html`) and the [Javadoc](#) (`../apidocs/1.2/index.html`)