

# RepositoryScenarios

## Scenarios for Repository Usage

### Introduction

Within the world of Java-based CMS, "repository integration" according to JSR-170 is a hot issue. A JSR is a Java specification request, established by the Java community. JSR-170 is a specification for a Java Content Repository API (shortform: JCR). See <http://www.jcp.org/en/jsr/detail?id=170> for all the details.

So, a natural question is: should the Lenya CMS become JSR-170 compliant ? Critics might say, of course, it needs to be because of buzzword-compliance. This page tries to explain scenarios why JSR-170 is a worthwhile effort, beyond the mere marketing buzzword.

### Using Lenya for CMS but something else for the live site

(copied from [PaulAnderson](#)'s email, thx Paul) We have Lenya's Cocoon-based authoring environment running on a single server (low criticality) on our internal network. Text items are rsynched from the content (live) directory to clustered Tomcat instances in the DMZ. When the JSP pages presenting the text items are served, image links for CMS binary items are XML transformed by XSLT in the JSP's. The content (authoring) directory is rsynched to the test and development machines so they always see the latest content. Binary items are rsynched from resources to the clustered Apache instances' content directory. We use Lenya as a backend CMS, for maintaining text and image items that are included in a JSP-based, XSLT-capable site. Unfortunately having no repository API (and we're not using Cocoon for presentation) means we can only specify the ID of a content item for display in a JSP - we can't do advanced selects by metadata or anything like that, unless we look more deeply into Lucene indexing or Cocoon web services. I hope this will be possible in Lenya in future if it switches to a JSR170 repository. (Also we'd prefer to be able to use WebDAV for maintenance and lose the dependency on Kupu - plus be able to maintain non-XHTML files easily).

### Using something else for initial Web site creation and then Lenya for Web site maintenance

Consider a Web site modeling tool, helping people to design their Web site according to whatever methodology. This tool may consist of various drawing tools, say one to specify which topics your Web site deals with and how they are related, another tool allows you to specify your site navigation (site tree) according to these topics, etc. The output of this modeling process is specified ... well, how exactly should it be specified ? Probably each modeling tool has its own format for specifying this information. But then what ?

Wouldn't it be cool if such a modeling environment wrote its output into a JSR-170 compliant repository ? The team who initially designed the site this way could then tell the editing team: the initial site is ready, you can now add contents and maintain it within Lenya, because Lenya is now also JSR-170 compliant !

(I (WolfgangKaltz) am currently involved in such a Web site modeling research project; I am hoping to steer this project towards it outputting its results into a JSR-170 compliant repository. Ultimately, I hope then that the output can be immediately read by Lenya, and the normal site maintenance proceeds from there, within Lenya)