CMS Application Reference Guide

# CMSe – Editorial

## Site Creation

* Start by editing slepeweb-init.xls.
* With CMSe running, browse to /cms/setup?file=slepeweb-init.xls
* Modify new row in site table – check 'secured' flag. A value of 1 means users need to login to access the site.
* Add row(s) into 'role' table, granting at least 'editor' role to users who need it.
* Add row(s) to 'access' table, granting 'w' access to 'editor' role.
* Add row to 'config' table to specify media storage mode. The property name is 'media\_storage\_mode', and the value can be either 'db' (database) or 'fs' (file system).
* Add row to 'host' table for server type 'staging'. More hosts for this site can be added later.
* Add line to /etc/hosts for this hosts, pointing to localhost.
* Re-start CMSe, to force cached data to be refreshed.

## Form field validation

A text field value (ie input type=”text/textarea”) can be constrained by validation rules. This is achieved by storing the corresponding regular expression in the 'validation' column of the 'field' table. If the stored expression contains backslashes, they must NOT appear as doubles, but singles. The expression needs to work in both javascript (front-end) and java (back-end), since the two languages might exhibit different results.

# CMSd - Site Delivery

## Request cycle

Each request is first handled by Spring's DispatcherServlet (see web.xml). The root mapping in PageController (/) uses CmsDeliveryServlet to deal with the request. For page requests (ie not images, etc), CmsDeliveryServlet identifies the item at the specified url, and forwards the request to a url that begins with '/spring'. The PageController then deals with building the page and specifying the rendering jsp.

# Common

## Access control

Both the editorial and site delivery apps respect the access control rules stored in the access table. In the case of CMSe, access to the application is controlled by LoginFilter. This checks for the existence of a session property, which is only placed in the session after successful login. In the case of CMSd, item visibility is checked by CmsDeliveryServlet, and if the rules deny access, then the user is redirected to the login page.

The Item bean has 'readAccess' and 'writeAccess' properties. The CMSe app currently sets the 'writeAccess' property when an item in the site is selected for editing, and although the editor can always see the item in the browser navigation, he is prevented from editing it if he doesn't have write access.

It would seem that currently, these properties are not being used in the CMSd app.