

Dashboard
Plug-in Guide
trunk-SNAPSHOT

Table of Contents

1 Introduction	4
2 Installation	5
2.1 Conventions	5
2.2 Installing Dashboard	5
2.3 Verifying the Installation	7
3 Database scripts	8
3.1 Example	8
4 Configuration	9
4.1 Enabling Moderation of Your Content	9
4.2 Enabling Preview of Moderated Images in Dashboard	10
4.3 Configuring the Dashboard Section Parameters	10
5 Logging into the Dashboard	11
5.1 Dashboard Security	11
6 Moderating Content	12
6.1 Using the Moderation States	12
6.2 Available Moderation States	13
6.3 Forum Moderation Schemes	13
7 Administering Users	14
7.1 Searching Users	14
7.2 The User Profile View	14
7.3 Banning and Unbanning Users	14
7.4 Assigning Roles to the Users	16
8 Community-Specific Features	17
8.1 Moderating Flagged Content	17
8.2 Tag Administration	18
8.2.1 Edit Tag	18
8.2.2 Delete Tag	19
8.2.3 Merge Tags	19
8.2.4 Search Tags	20
8.3 Permission Administration	20
9 Forum-Specific Features	23
9.1 Forum Filtering	23
9.2 Using Stop Words	23
9.2.1 Adding a Stop Word	24

9.2.2 Searching a Stop Word	24
9.2.3 Deleting a Stop Word	24
10 Protecting the Dashboard	25

1 Introduction

The Dashboard plug-in provides a web interface for moderating user-generated content, such as blog posts, article comments, pictures and videos. The Dashboard also allows a publication administrator to managing user access rights, stop words, tags and so on.

The Dashboard plug-in can be used at two types of Content Engine installation:

- Installations where both the Forum plug-in and Viz Community Expansion are installed
- Installations where only the Forum plug-in is installed

The Dashboard features that are specific to Viz Community Expansion are described in [chapter 8](#).

2 Installation

The following preconditions must be met before you can install Dashboard trunk-SNAPSHOT:

- The Content Engine and Escenic assembly tool have been installed as described in the **Escenic Content Engine Installation Guide** and are in working order.
- You have the required plug-in distribution file `dashboard-trunk-SNAPSHOT.zip`.

2.1 Conventions

The instructions in the following section assume that you have a standard Content Engine installation, as described in the **Escenic Content Engine Installation Guide** for releases 5.3.0 and above. `escenic-home` is used to refer to the `/opt/escenic` folder under which both the Content Engine itself and all plug-ins are installed.

The Content Engine and the software it depends on may be installed on one or several host machines depending on the type of installation required. In order to unambiguously identify the machines on which various installation actions must be carried out, the **Escenic Content Engine Installation Guide** defines a set of special host names that are used throughout the manual.

Some of these names are also used here:

assembly-host

The machine used to assemble the various Content Engine components into a enterprise archive or .EAR file.

database-host

The machine used to host the database server.

engine-host

The machine(s) used to host application servers and Content Engine instances.

The host names always appear in a bold typeface. If you are installing everything on one host you can, of course, ignore them: you can just do everything on the same machine. If you are creating a larger multi-host installation, then they should help ensure that you do things in the right places.

2.2 Installing Dashboard

Installing Dashboard involves the following steps:

1. Log in as **escenic** on your **assembly-host**.
2. Download the Dashboard distribution from the Escenic Technet web site (<http://technet.escenic.com>). If you have a multi-host installation with shared folders as described in the **Escenic Content Engine Installation Guide**, then it is a good idea to download the distribution to your shared `/mnt/download` folder:

```
$ cd /mnt/download
$ wget http://user:password@technet.escenic.com/downloads/release/54/dashboard-
trunk-SNAPSHOT.zip
```

Otherwise, download it to some temporary location of your choice.

3. If the folder `/opt/escenic/plugins` does not already exist, create it:

```
$ mkdir /opt/escenic/plugins
```

4. Unpack the Dashboard distribution file:

```
$ cd /opt/escenic/plugins
$ unzip /mnt/download/dashboard-trunk-SNAPSHOT.zip
```

This will result in the creation of an `/opt/escenic/plugins/dashboard-<version>` folder.

5. Log in as **escenic** on your **assembly-host**.
6. The first time you re-assemble your EAR with the Dashboard include, you need to create a dashboard publication resource for the Assembly Tool to use. Typically, this would mean:

```
$ vi /opt/escenic/assemblytool/publications/dashboard.properties
```

In this file you enter

```
name: dashboard
source-war: /opt/escenic/plugins/dashboard-trunk-SNAPSHOT/wars/dashboard-
webapp.war
context-root: dashboard
```

The Assembly Tool can now build your dashboard alongside the other web applications.

7. Run the **ece** script to re-assemble your Content Engine applications

```
$ ece assemble
```

This generates an EAR file (`/var/cache/escenic/engine.ear`) that you can deploy on all your **engine-hosts**.

8. If you have a single-host installation, then skip this step.

On each **engine-host**, copy `/var/cache/escenic/engine.ear` from the **assembly-host**. If you have installed an SSH server on the **assembly-host** and SSH clients on your **engine-hosts**, then you can do this as follows:

```
$ scp -r escenic@assembly-host-ip-address:/var/cache/escenic/engine.ear /var/
cache/escenic/
```

where `assembly-host-ip-address` is the host name or IP address of your **assembly-host**.

9. On each **engine-host**, deploy the EAR file and restart the Content Engine by entering:

```
$ ece deploy
$ ece restart
```

10. The last thing needed for installing Dashboard, is to create a publication that matches the dashboard web application you assembled above.

Go to the **escenic-admin** web application and select **New pubs**.

11. Then choose the `dashboard/wars/dashboard-webapp-trunk-SNAPSHOT.war` as the publication resource. **escenic-admin** should find all the resources it needs for the dashboard.
12. Choose a name for the dashboard web app, we recommend simply using **dashboard** as the name. The name must be same as the one you choosed for the dashboard publication before assembling the EAR file.

You should now have a successfully installed Dashboard. Continue reading the next section on how to verify your installation.

2.3 Verifying the Installation

To verify the status of the Dashboard plug-in, open the Escenic Admin web application (usually located at <http://server/escenic-admin>) and click on **View installed plugins**. The status of all currently installed plug-ins is shown here, and indicated as follows:



The plug-in is correctly installed.



The plug-in is not correctly installed.

So if the Dashboard plug-in is correctly installed, you should see something like this in the displayed plug-in list:

Home List pubs New pubs	Dashboard		trunk-SNAPSHOT	Dashboard	ECE Dashboard plugin
---	------------------	---	----------------	-----------	----------------------

3 Database scripts

Dashboard comes with database scripts located in the `misc/database/{mysql,oracle}` folder of the bundle.

1. `constants.sql`

These scripts should be installed into the same database schema as the ECE is using, using the same DB user and password.

Please see the ECE install documentation on how to set up the database and ECE database schema.

3.1 Example

If you have installed the software as described in [section 2.2](#), you may follow this example on how to install all the VCE database scripts on a `mysql` database running on a server called `dbhost` with the schema `ecedb`, user name `eceuser` and password `ecepassword`:

```
$ for e1 in constants; do
  for ele in `find -L /opt/escenic/community-engine/ -name "$e1.sql" | grep mysql`; do
    mysql -u eceuser -pecepassword -h dbhost ecedb < $ele
  done
done
```

The equivalent for Oracle DB would be:

```
$ for e1 in constants; do
  for ele in `find -L /opt/escenic/community-engine/ -name "$e1.sql" | grep oracle`;
do
  sqlplus eceuser/ecepassword @$ele
done
done
```

4 Configuration

Configuring the Dashboard involves the following steps:

- Choosing the content you wish to be moderated
- Choosing how you want the content to be displayed in the Dashboard

4.1 Enabling Moderation of Your Content

To select what content is to be moderated:

1. Download your publication's **content-type** resource. You can do this either using the **escenic-admin** web application, or using a command line tool such as **wget**. For a publication called **mysite**, for example, you could do this as follows:

```
$ wget http://host:port/escenic-admin/publication-resources/mysite/escenic/
content-type
```

2. Open the **content-type** resource in an editor of your choice.
3. Add a **com.escenic.community.dashboard.moderate** parameter and a **dashboard-moderator-fields** field group reference to each content type you wish to be moderated. For example:

```
<content-type name="my-content-type-that-we-want-to-moderate">
  [..]
  <parameter name="com.escenic.community.dashboard.moderate" value="true"/>
  [..]
  <panel ..>
    [..]
    <ref-field-group name="dashboard-moderator-fields"/>
  </panel>
</content-type>
```

4. Add the **com.escenic.community.dashboard.showField** parameter to each field definition you wish to be shown in Dashboard. For example:

```
<field mime-type="text/plain" type="basic" name="my-field-name">
  [..]
  <parameter name="com.escenic.community.dashboard.showField" value="true"/>
  [..]
</field>
```

Note that, title fields are always visible and editable in Dashboard.

5. Define the **dashboard-moderator-fields** group as follows:

```
<field-group name="dashboard-moderator-fields">
  <field name="moderation-state" type="enumeration">
    <ui:label>Moderator Status</ui:label>
    <enumeration value="approved"/>
    <enumeration value="not-reviewed"/>
    <enumeration value="pending"/>
    <enumeration value="rejected"/>
    <enumeration value="reviewed"/>
  </field>
</field-group>
```

6. POST the updated **content-type** resource back to its correct location. You can either do this using **escenic-admin**, or as follows using **wget**:

```
$ wget --post-file=content-type \
  -S \
  -O - \
  http://host:port/escenic-admin/publication-resources/mysite/escenic/content-
type
```

4.2 Enabling Preview of Moderated Images in Dashboard

The image content type definition must have a representation field **dashboardreps** containing two representations named **thumb** and **preview** along with the parameter and fields discussed above in order to show the moderated images in Dashboard. Define the representation field like this:

```
<field name="dashboardreps" type="basic" mime-type="application/json">
  <ui:hidden/>
  <representations xmlns="http://xmlns.escenic.com/2009/representations"
type="image-versions">
    <representation name="preview">
      <output width="400" height="300"/>
      <crop/>
      <resize/>
    </representation>
    <representation name="thumb">
      <output width="100" height="100"/>
      <crop/>
      <resize/>
    </representation>
  </representations>
</field>
```

4.3 Configuring the Dashboard Section Parameters

You can control how moderated content is displayed in the Dashboard by adding section parameters to sections using Web Studio. Note that in order to be able to add section parameters to a section, you must have administrator rights for the section. For details of how to set section parameters, see the **Escenic Publication Administrator Guide**.

Here are some of the parameters you can set:

content.length.max

Determines the maximum field length (in characters) in the Dashboard's content moderation view. If any field in a content item is longer than **content.length.max**, then it is truncated, and a link to the full content item on the live site is displayed.

5 Logging into the Dashboard

You log into Dashboard by accessing the URL of the Dashboard web application. Normally this webapp is called **dashboard** and will be available under this URL: <http://my-editorial-host/dashboard/> where **my-editorial-host** is the host where your editorial staff is connecting their Escenic Content Studio to.

If you wish to allow external users to log into the Dashboard, the **dashboard** web application must of course be deployed on the hosts facing the web (the presentation servers).

On the login page, it is important that you select the correct publication to log in to, as well as your user name and password. You can always use the publication admin user to log in. Alternatively, you can assign the roles defined in [Dashboard Security \(section 5.1\)](#) to give other users sufficient privileges to use the Dashboard, either for content moderation only or for using all the Dashboard features.



The screenshot shows a login form with three input fields and a button. The 'User-name' field contains 'community_admin', the 'Password' field contains six asterisks, and the 'Publication' dropdown menu is set to 'community'. A 'Login' button is located to the right of the 'Publication' field.

5.1 Dashboard Security

In order to be able to access the Dashboard web application, a user must have one of the following roles:

dashboard-administrator

A user with this role has full access to all Dashboard functionality. (Note that the publication administrator is automatically granted this role.)

content-moderator

By default, a user with this role only has access to the Dashboard's **Content Moderation** functionality.

Both normal editorial users and community users (if the Viz Community Expansion is installed) may log into the Dashboard, as long as they have been assigned one of these roles.

These roles are assigned to users using Web Studio in the usual way. For details, see the **Escenic Content Engine Publication Administrator Guide**.

Unless you actually need to allow community users to moderate parts of your web site, we recommend disallowing access to the Dashboard web application for non-editorial users.

6 Moderating Content

The Dashboard plug-in allows editorial staff to quickly deal with hundreds of user generated content items. It works by using a **moderation-state** field that must be added as a hidden field to all content types that are to be moderated (see [chapter 6](#)). Moderators, when they accept or reject a content item, or set it to **pending**, change the value assigned to this field. Template developers can then access this field and determine how to display (or not display) a content item depending on its moderation state. This gives template developers a great deal of flexibility in how moderation is actually implemented in their publications.

A typical moderation scenario is:

1. User Lisa writes a blog entry about how much she dislikes life, the universe and everything.
2. Editor John logs into the Dashboard, searches for all blog entries written today.
3. John sees Lisa's blog entry and finds it a bit too offensive. He sets the blog entry's moderation state to **rejected**.
4. Lisa sees on her profile page that her latest blog entry has been rejected, so she edits it to make it more acceptable.
5. John sees that the rejected entry has been changed and reads it again. Now it's OK, so he sets the moderation state to **accepted**.
6. All web site users and visitors can now read Lisa's edited blog entry again.

This is only one of several possible ways moderation can work. It can, for example be set up in a more restrictive way so that nothing is published until the moderator has accepted it (this is called **pre-moderation**).

In addition to the **moderation-state** field, the Dashboard plug-in provides a user interface that moderator can use to set the state stored in the field.

Some changes to the **moderation-state** field trigger a corresponding change in a content item's **state** property. For instance, marking a content item as **pending** or **rejected** automatically sets the content item's **state** property to Submitted so that is no longer visible in normal listings, such as the lists returned by the **article:list** tag (unless **onlyLive** is set to **false**).

Precisely how moderation works at any particular installation depends on how publication templates make use of these states to choose what get displayed when. The following sections describe how the template developer can access the moderation states.

6.1 Using the Moderation States

If a content item's **moderation-state** is set to either **rejected** or **pending** then it cannot be retrieved by the **article:list** tag unless **onlyLive** is set to **false**. Usually you would want rejected content to be viewable by the person who wrote it (on his/her profile page most likely), but not by other users browsing the blog writer's page.

You can read a content item's moderation state as follows:

```
| <dashboard:moderationLog
```

```
id="moderationLog"
articleId="{currentArticle.id}"
moderationType="edited" />
<core:if test="{fn:length(moderationLog) > 0}">
  This blog has been edited by editorial staff,
  {moderationLog[0].userName}</a> on
  {moderationLog[0].creationDate}
</core:if>
```

6.2 Available Moderation States

Currently, the following moderation states are available:

- approved
- edited
- not-reviewed
- pending
- rejected
- reviewed

To avoid typing errors in your code it's a good idea to use constants for these states. They are available in `com.escenic.community.DashboardConstants`, where the list of available states will always be kept up to date.

6.3 Forum Moderation Schemes

Moderation of forum postings and article comments managed by the Forum plug-in is controlled by **moderation schemes**. The moderation scheme that selected determines what initial moderation stated is assigned to a forum posting or comment when it is created. Currently the Forum plug-in supports two moderation schemes:

Pre-moderation

Postings are initially assigned a state of **pending**?. This means they will not be published until they have been approved by a moderator.

Post-moderation

Postings are initially assigned a state of **not-reviewed**. This means they are published immediately, but may be subsequently withdrawn by a moderator.

For more information about the set-up required to enable moderation schemes in Forum content types, see the **Forum Plug-in Guide**.

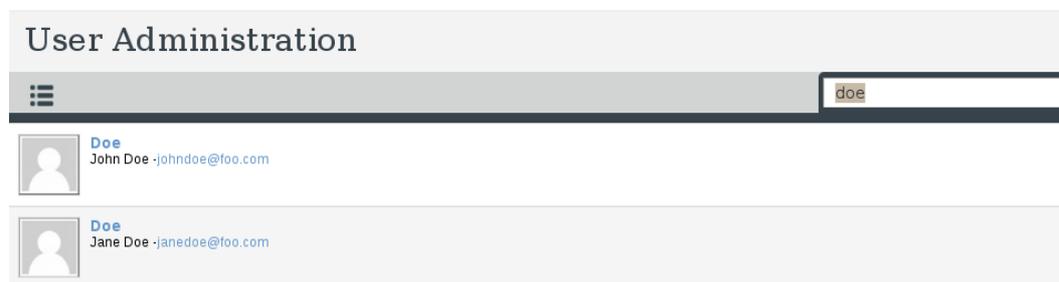
Moderation schemes are not available for other types of moderated content, such as user-written blogs.

7 Administering Users

Using the **User Administration** module, the Dashboard administrator can find the users. Here are few features in this module that can be used to administer the users.

7.1 Searching Users

The **User Administration** page shows a list of users. The list initially loads few users but it loads more users in the list as the administrator scrolls the page down. A search box on the top right can be used to search users. The search is performed on username, firstname, surname and the email address of the user.

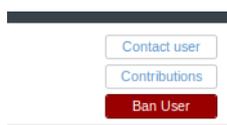


7.2 The User Profile View

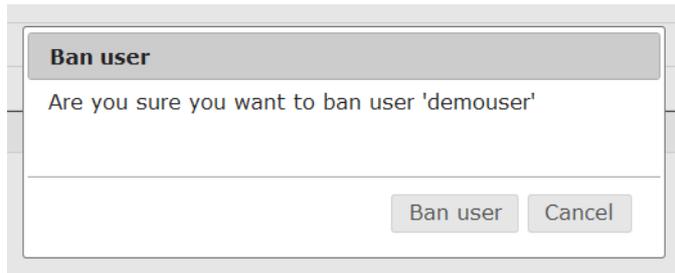
Clicking a username in the user list will open a user profile page. This page displays various information about the user. It provides a link to the contributed content of the user and provides the user's contact email address. It is also possible to ban or unban a user from this interface.

7.3 Banning and Unbanning Users

The Dashboard plug-in's moderation feature allowed the moderator to reject the specific blog entry. However, if the community user repeatedly writes offensive entries, the administrator can ban him from his user profile page. In the user profile page, the link **Ban User** is used for this purpose. The user profile page can also be accessed from **Content Moderation** just by clicking the user name.



If you click on this **Ban User** button, a confirmation dialog is displayed:

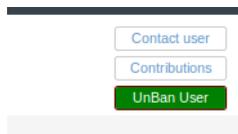


If you again select **Ban user**, then the user is banned. The consequences of banning are:

- The user is no longer able to log into the community site. Any attempt to log in will result in the display of an error message.
- If the configuration file `/com/escenic/community/messaging/BanReporter.properties` has been configured correctly, an e-mail is sent to the banned user. This configuration file can be used to define the sender, subject and content of the notification message. For example

```
$class=com.escenic.community.messaging.BanReporter  
fromUser=username@domain.com  
subject=User Banned Notification  
body=This mail is to notify you that you have been banned.
```

When you have banned a user, the **Ban user** button displayed below the content item is replaced by an **Unban user** button, which you can use to restore the user's access rights:

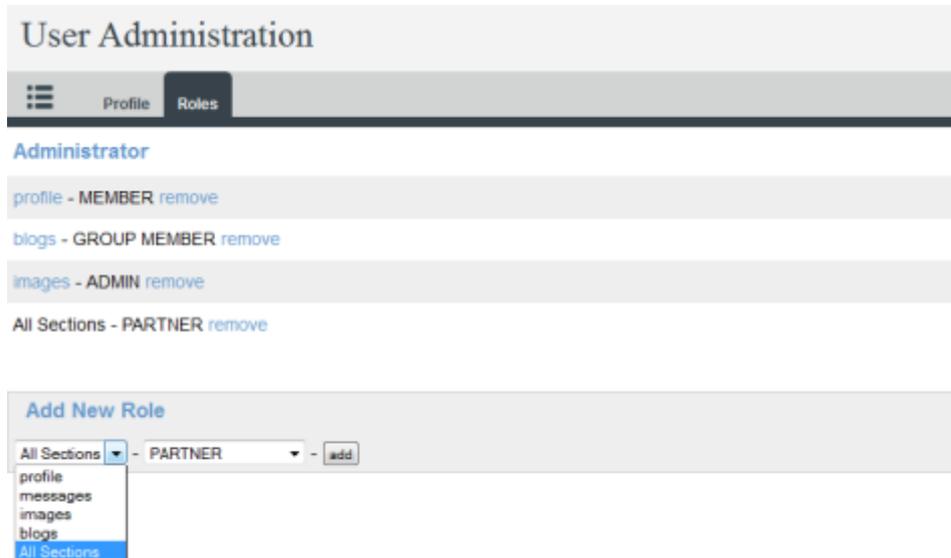


You have to confirm this action in the same way as when banning a user:



7.4 Assigning Roles to the Users

The **Roles** tab in the user profile page shows a list of roles that are already assigned to the user.



Administrator can add roles to the user for specific sections. To add a role, the administrator has to choose the role and section from the drop down list and click on **Add** button. There is also **All section** in the section list on which the administrator can easily assign the user role. This way, the user enjoys the role all through the publication.

Please note that the section list in this interface only contains the user content sections and the group sections. In addition, to appear in this list the user content sections have to be specified using section parameters in the root section of the publication. Please have a look at the **Viz Community Expansion Developer Guide** to see how to configure user content section.

Administrator can also remove a role from the user by clicking **remove** link.

8 Community-Specific Features

This chapter describes the Dashboard plug-in's community-specific features:

- Moderating flagged content
- Tag Administration

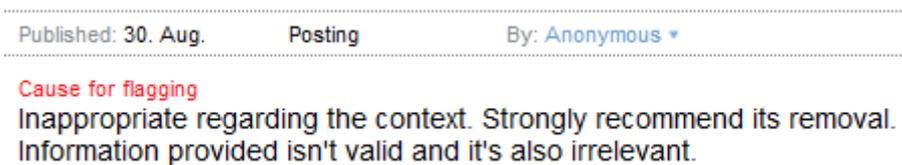
8.1 Moderating Flagged Content

Community sites allow users to flag content they find offensive. Any content that is flagged in this way:

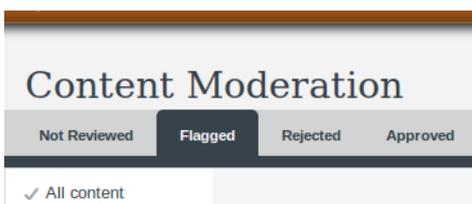
- Is immediately withdrawn from view for ordinary users
- Is visually flagged in the **All content** tab of the Dashboard's **Content Moderation** view:



- Shows any complaints as cause for flagging in the **Flagged** tab of the Dashboard's **Content Moderation** view:

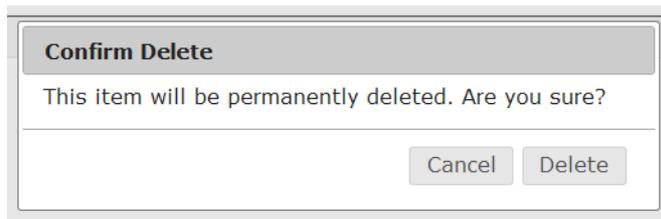


- Is added to the **Flagged** tab of the Dashboard's **Content Moderation** view:



The moderator can either:

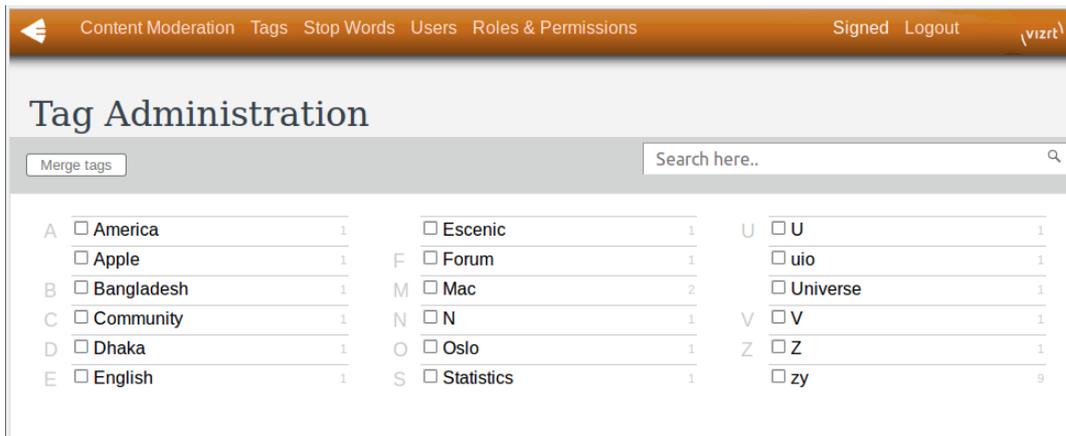
- Restore a flagged item by clicking **Restore** if it does not breach editorial policy, or
- Delete the item by clicking **Reject**. This action requires confirmation:



Restoring an item removes its flag and makes it visible again. However, its moderation history can be revealed by mouse-hovering on the **View Moderation Log** in the bottom right-hand corner of the flagged item:



8.2 Tag Administration

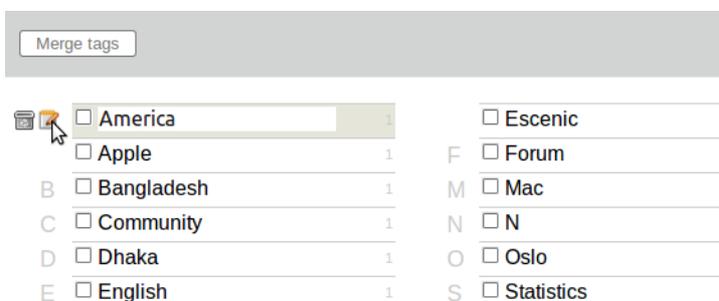


Dashboard allows the moderators to administer the tags added by the community users. In **Tag Administration** module, a moderator can perform the following operations:

- Edit Tag
- Delete Tag
- Merge Tags
- Search Tags

8.2.1 Edit Tag

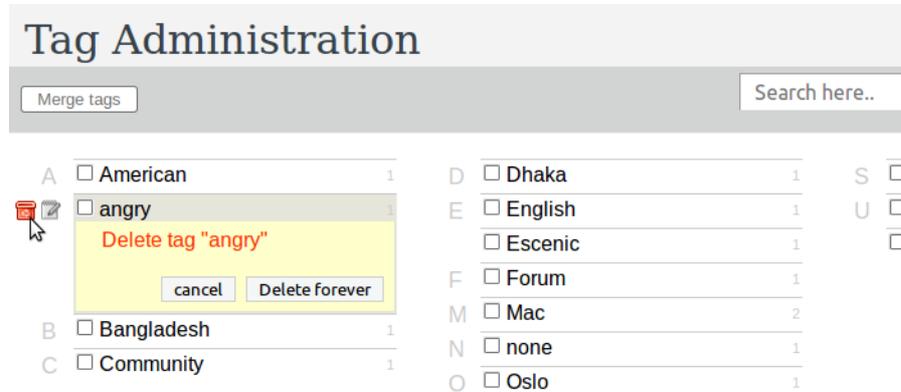
When a tag is hovered, a tools menu shows up on the left of that tag.



Clicking on the edit icon will put the tag in edit mode. You can then edit it. And when you remove your mouse from the tag it's saved.

8.2.2 Delete Tag

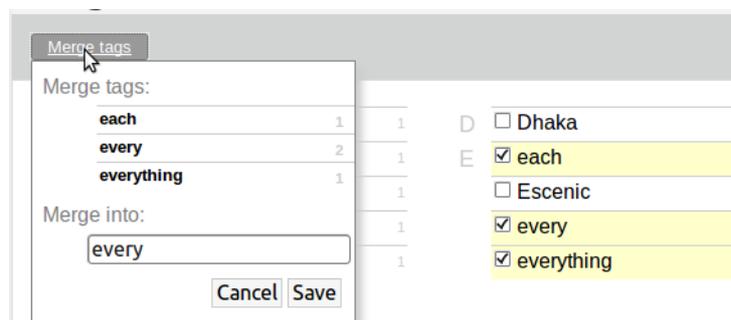
The tools menu that shows up while hovering the tag contains a delete icon. Clicking on that icon will prompt for confirmation of tag deletion.



Clicking on **Delete forever** will delete the tag and its references to content items permanently.

8.2.3 Merge Tags

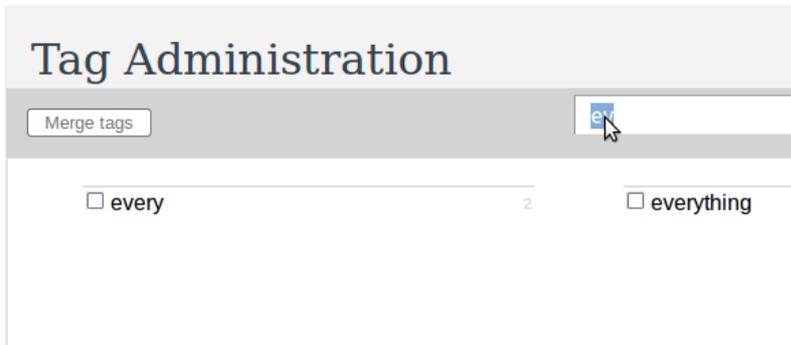
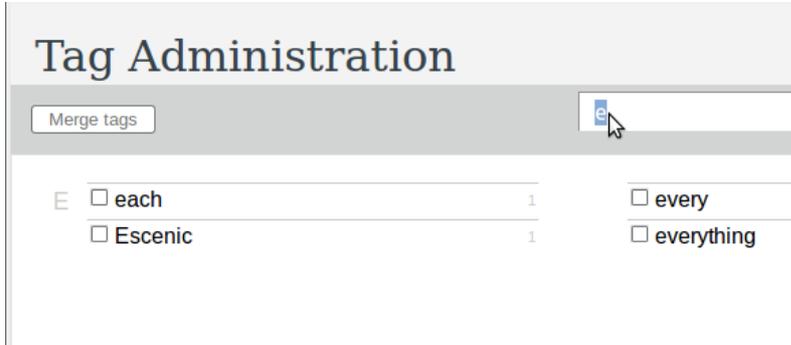
To merge tags, check the checkboxes of the tags you want to merge. Now, click on the **Merge tags** button. Then you'll see a menu showing all the tags you selected and an edit box holding the most used tag out of the tags you selected.



Clicking on the cancel button will cancel the operation. Clicking on the save button will merge all the tags in the merge menu into the one specified in the edit box.

8.2.4 Search Tags

Type in the search box on the right top of Tag Administration and for each keypress only the tags matching your search text will be shown.



8.3 Permission Administration

The **Permission Administration** module lets the Dashboard administrator configure the the roles and the mapping with roles and permissions using a matrix.

- The Administrator can add roles by clicking the **Add** button. A text box will appear at the bottom of the **Role** column of the role-permission matrix. Now, the administrator can input the name of the role and press **Enter** to save it.

Permission Administration

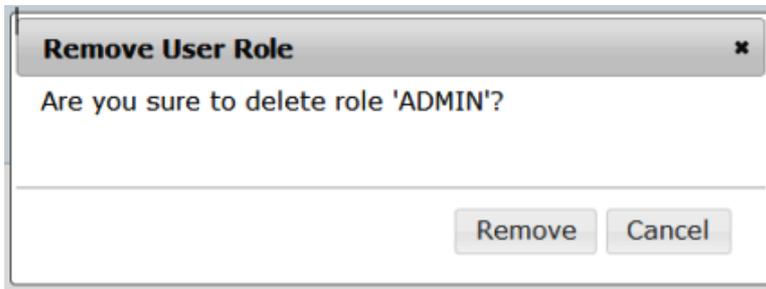
Add

Role \ Permissions	flag	inviteToGroup	moderateGroupUser	rate	
ADMIN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remove
GROUP MEMBER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	remove
GROUP MODERATOR	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remove
GROUP OWNER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	remove
OWNER <input style="width: 80px;" type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Cancel

- The Administrator can assign permissions to an existing role as well as a newly created role. Permissions can be assigned by checking the appropriate checkbox in **Role/permissions** matrix. In the same manner, it is possible remove the permission from the role simply by removing the check from the checkbox.

Role \ Permissions	flag	inviteToGroup	moderateGroupUser	rate	
ADMIN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remove
GROUP MEMBER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	remove
GROUP MODERATOR	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	remove
GROUP OWNER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	remove

- A role can be removed by clicking **remove** link in the same row of the role. A confirmation dialogue box appears when the link is clicked. If the administrator confirms, that role is deleted.



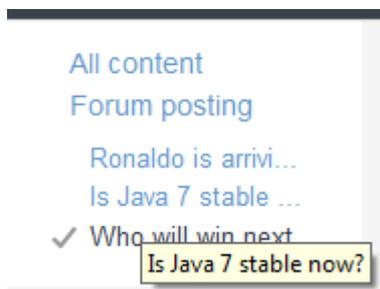
Please note that, if the role is already assigned to any user, there will be a second confirmation box before deleting the role showing that the role is already assigned. Also note that the users may not be able to use some community site features after deleting such role.

9 Forum-Specific Features

This chapter describes the Dashboard plug-in's Forum specific features. These features are of course available also when Community Expansion is installed as this plug-in depends on Forum, but they only apply to the Forum specific components (discussion forums and postings) and not the Community entities such as blogs and uploaded images.

9.1 Forum Filtering

Forums are listed on left panel of dashboard. Clicking on a forum shows all the postings in it.



9.2 Using Stop Words

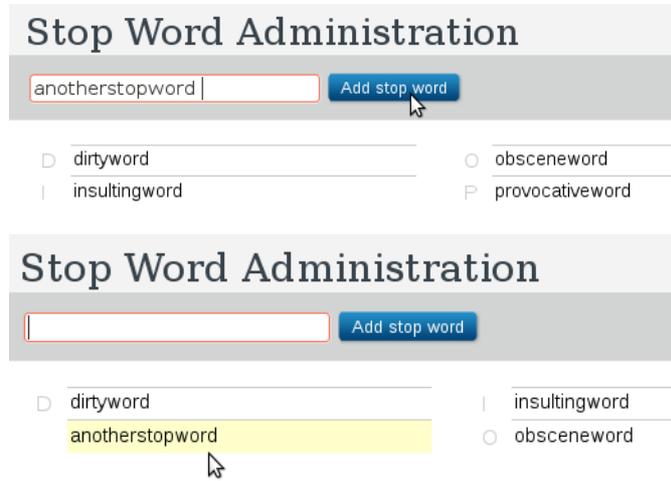
One of the Dashboard modules allows you to edit the stop words on your web site. If a user posts a forum posting entry or article comment with a word that's in this list, the posting will be rejected immediately and will not enter the system.



Stop Word Administration module allows the moderator to add, search, delete a stop word.

9.2.1 Adding a Stop Word

To add a stop word, enter one or multiple stop words separated by spaces, and press the **Add stop word** button. Newly added stop words will show up in the list with yellow background.



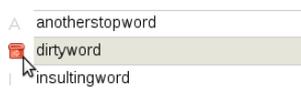
9.2.2 Searching a Stop Word

Typing in the search box will scope down the stop word list shown with prefix match.



9.2.3 Deleting a Stop Word

Hovering a stop word in the list will show a delete button on the left of the stop word. Clicking on the button will delete the stop word immediately.



10 Protecting the Dashboard

The current version of Dashboard will allow any logged in user to moderate all content. Thus, remember to protect the Dashboard from the outside world by running it only on the editorial (publication) server or blocking it from all other networks than your local one. This access control can be done in your web server (e.g. Apache) or cache server (e.g. Varnish).

Here is an example how to restrict access to the Web Studio (normally called "escenic"), Phoenix (normally called "escenic-admin") and Dashboard (normally called "dashboard") webapps using the Varnish cache server:

```
acl mynetwork {
    "192.0.2.0"/24;
}

sub vcl_recv {
    [..]
    if (!client.ip ~ mynetwork &&
        (req.url ~ "^/escenic" ||
         req.url ~ "^/escenic-admin" ||
         req.url ~ "^/dashboard")) {
        error 405 "Not allowed.";
    }
    [..]
}
```