

# Roller Group Blogging Design & Plan – DRAFT1

**Author:** Dave Johnson, Sun Microsystems, **Date:** April 13, 2005, **Target Release:** Roller 2.0

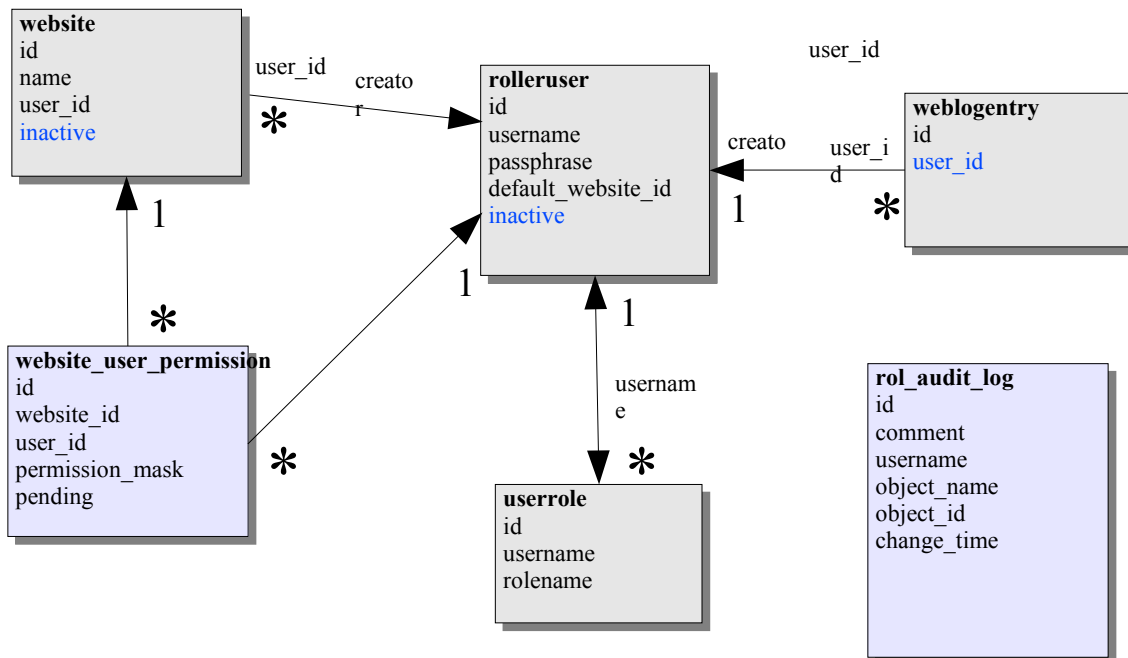
This document is the design document for group blogging in the Roller blog server. In simple terms, the design is to add a many-to-many relationship between the Website and User, a simple workflow for Weblog Entries, and the persistent objects and the UI to support the requirements. One design goal is to have zero-impact on single-user bloggers: we don't want the group blogging changes to require single-user bloggers to do any extra work or even to be aware of the group blogging workflow.

The design is divided into database schema changes, backend changes, and frontend interface.

## 1.0 Database Schema Changes

- Many-to-many Users to Websites: To allow for a many-to-many relationship between Websites and Users, we will add new table `website_user_permissions`.
- User Permissions. User permission within a Website will be defined by a 16-bit bitmask (001 BLOG\_LIMITED, 010 BLOG\_AUTHOR, 100 BLOG\_ADMIN).
- Website Owner. A many-to-one relationship between Websites and owner Users will be maintained via the existing relationship between the `website` and `roller_user` tables.
- Weblog Entry Attribution. A new creator field will be added to the `weblogentry` table that specifies the username of the original creator of the entry.
- Inactive Users and Websites. New inactive field will be added to the `website` and `roller_user` tables.
- Weblog entry auditing. A new `rol_audit_log` table will be added to record each weblog entry change.

**Figure 1: Group blogging data model changes**



**Open issue:** replace `userrole` table with `website_user_permission` page coupled with our own authentication system?



## 1.1 New Configuration Properties

These could be added to the in `roller_config` table or in a new configuration system (if available)

- New User Policy (user registration, admin only registration)
- Website creation policy (user creates many, user creates one, admin creates)
- Website deletion policy (user delete allowed, admin only delete)
- Blog Author policy (collaboration or single-author)

## 2.0 Backend changes

### 2.1 Interface change (`org.roller.model`)

Most already interfaces use `Website` rather than `User` and therefore do not need to be changed.

- `IndexManager` – `removeUserIndex()` should become `removeWebsiteIndex()`
- `UserManager`
  - `addUser()` - will no longer create a website for a user, it will just create a user.
  - `getWebsite(user)` - will no longer exist
  - `removeUserWebsites(user)` - will no longer exist
  - `getOwnedWebsites(user)` - (new) Get all websites owned by specified user
  - `getMemberWebsites(user)` - (new) Get all websites in which user is member
  - `getWebsiteInvitations(user)` - (new) get all pending `Website` invitations for a user.
  - `getWebsiteInvitations(website)` - (new) get all pending `Website` invitations for a website
  - `getUserPermissions(user, website)` - (new) Get `WebsitePermissionData` object for user/website
  - `createWebsite(user, args...)` - (new) Create new website owned by user
  - `createWebsitePermission()` - create new `WebsitePermissionData`
  - `storeWebsitePermission()` - save or update `WebsitePermissionData`
  - `removeWebsitePermission()` - remove `WebsitePermissionData`
- `WeblogManager`
  - Methods that save, update, and remove weblog entries must record these actions in the audit table.
- `Audit Manager`
  - `createObjectAudit()` - create new audit record
  - `storeObjectAudit()` - save audit record

### 2.2 Persistent object changes (`org.roller.pojos`)

- `WebsitePermissionsData` (new) – this object is an association object that creates a many-to-many relationship between `WebsiteData` and `UserData`.



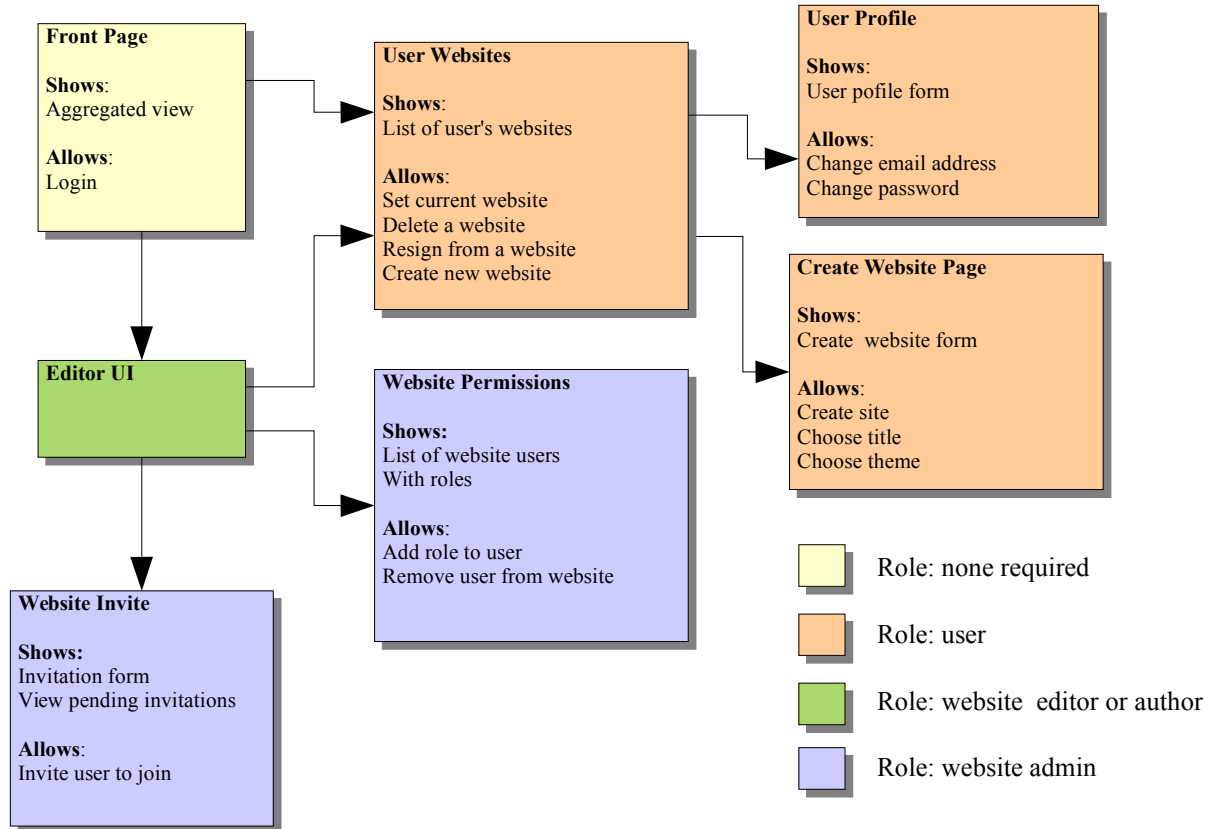
- ObjectAuditData (new) – This is a simple object with no associations.
- WebsiteData – **add inactive flag**
- WeblogEntryData – **add user\_id field, remove published field, add status field.**



### 3.0 User Interface design

Overview of new pages to be added. Each new page (and the changes to the weblog edit page) is discussed in the pages that follow.

**Figure 2: Overview of new UI to be added**





### 3.1 User profile

Allows user to edit his or her name, email address, password, etc. This is essentially the Website:User page, but moved to a different position in the menu.

Figure 3: User profile page

**My Profile**

Be sure to enter a valid email address or you will miss important system messages

Full Name	<input type="text"/>	?
Email Address	<input type="text"/>	?
Password	<input type="password"/>	?
Password (confirm)	<input type="password"/>	?
Locale	Albanian	?
Timezone	ACT - America/Eirunepe	?



### 3.2 User websites page

Allows user to view list of all of her websites, resign from a website, and create new websites (if permitted by policy). Pending Website membership invitations will be shown at the top of the page with Accept and Decline links or buttons.

Figure 4: User websites page

**Your Websites**

You've been invited to join the website [Kernel Space] do you: [accept](#) | [decline](#)

These are the websites of which you are a member. You may select one and it will become your *current website*. If you would like to end your membership in a Website, you may resign.

Title	Description	Roles	Select	Resign
Poodle Puppies United	Your dally source of canine foosball news	ADMIN EDITOR		
Frisbee Fool Festival	Team blogging the Ultimate Nationals	ADMIN EDITOR		
Blogveggie Fresh	Vegetarian's guide to blogs and wiki software	AUTHOR		
AJAX Kiddies	The six-to-eight year old's guide to XMLHttpRequest	AUTHOR		
Cure This!	Amateur psychiatry for fun and profit	ADMIN EDITOR		

**Your Options**

Edit your [User Profile](#)

Create a [New Website](#)

Actions:

- **Display:** populate page model with user's WebsitePermissionData objects and forward to JSP to display those objects in the table.
- **Accept invitation:** set the pending flag to false on the WebsitePermissionData object specified in the request. Notify the BLOG\_AUTHORS and BLOG\_ADMINS of the Website via email.
- **Decline invitation:** delete the WebsitePermissionsData object specified by the request, so that it is no longer a pending membership. Notify the BLOG\_AUTHORS and BLOG\_ADMINS of the Website via email.
- **Select current Website:** set the user's current Website in the Servlet session to the one specified by the request.
- **Resign from Website:** Remove the WebsitePermissionData object specified by the request. Notify the BLOG\_AUTHORS and BLOG\_ADMINS of the Website via email.

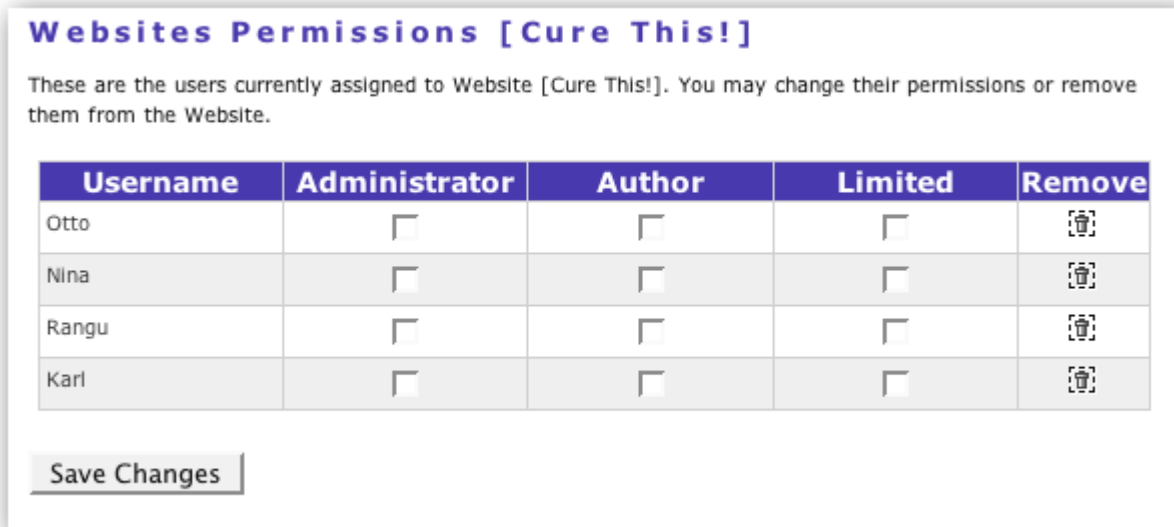
Note: is select current website really necessary since you can do that via the Select Current Website combo-box that is to be on every Editor/Admin UI page?



### 3.3 Website permissions page





Allows Website Administrator to grant and revoke permissions from users in a Website

**Figure 5: Website permissions page**



**Websites Permissions [Cure This!]**

These are the users currently assigned to Website [Cure This!]. You may change their permissions or remove them from the Website.

Username	Administrator	Author	Limited	Remove
Otto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Nina	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Rangu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Karl	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

#### Actions

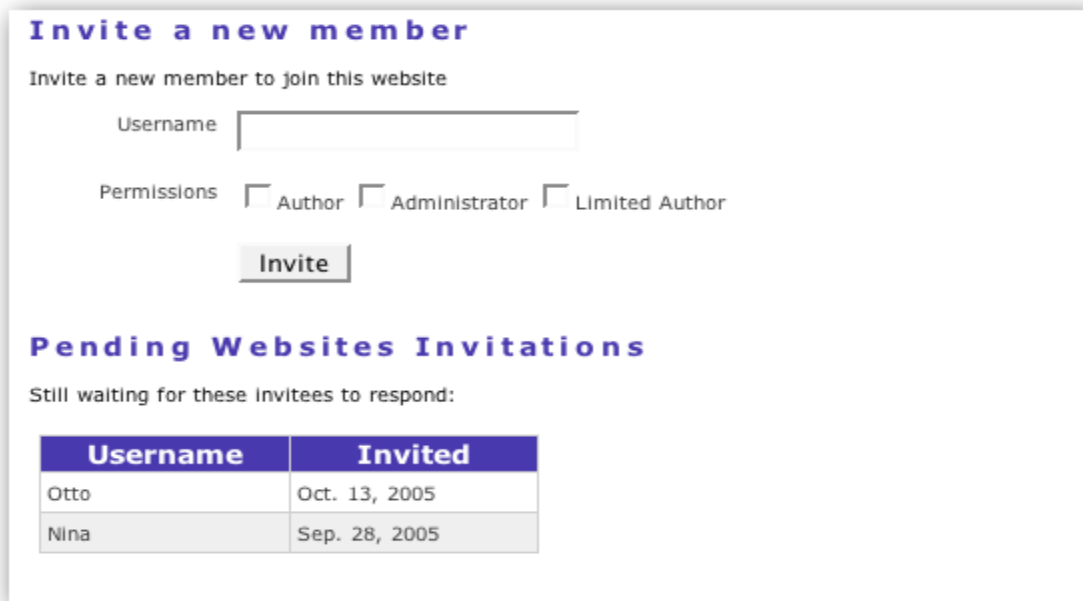
- **Display:** populate page model with website's `WebsitePermissionData` objects and forward to JSP for display.
- **Save changes:** save changes to `WebsitePermissionData` objects indicated by request.
- **Remove user:** delete specified user's `WebsitePermissionData` object and notify them via email of the change.



### 3.4 Invite new member page

Allow Website Administrator to invite a new member to join a Website.

**Figure 6: Website invite page**



**Invite a new member**

Invite a new member to join this website

Username

Permissions  Author  Administrator  Limited Author

**Pending Websites Invitations**

Still waiting for these invitees to respond:

Username	Invited
Otto	Oct. 13, 2005
Nina	Sep. 28, 2005

Actions:

- **Display:** populate page model with pending WebsitePermissionData objects for the Website and forward to JSP for display.
- **Invite:** use posted form data to new WebsitePermissionData for specified user, give it pending status.



### 3.5 Weblog Edit Page

The Weblog Edit page will have to change to take into account group blogging.

#### 3.5.1 Changes to Struts Action methods

The buttons shown will have to change.

- Users with BLOG\_LIMITED permission should not see a **Publish** button, instead they should see a **Submit for Review** button which will put their entry in the queue for review by a user with BLOG\_AUTHOR permission.
- Users with BLOG\_AUTHOR permission should see the **Publish** button and if the entry is in PENDING state, he should also see a **Return for Editing** button that will return the entry to DRAFT state for revision by those with BLOG\_LIMITED permission.

And the corresponding new actions will be added:

- Submit for review: save entry and set status to PENDING. Notify website's BLOG\_AUTHOR and BLOG\_EDITORS of the change via email.
- Return for editing: save entry and set status to DRAFT. Notify entry's creator of the change via email.

#### 3.5.2 Changes to side-bar list of entries

The lists of recent drafts and recent entries will have to change too:

- Users with BLOG\_LIMITED should only see the list of recent drafts
- Users with BLOG\_AUTHOR should see both lists

Those lists of recent drafts and entries are not really sufficient for situations where there are a large number of drafts to be reviewed. As a stretch goal perhaps we should introduce variants of the Weblog:Archive that list DRAFT and PENDING entries.



## 4.0 Development Plan

The purpose of this plan is to break the work down into tasks so we can make estimates, coordinate work, and arrive at a release date for group blogging. Estimates will be made in a separate document.

The plan is to first put UI unit tests in place (to complement our existing backend tests), make the database/backend changes, and then pick up the pieces. We'll know where done with the unit tests run and Roller works as it did before. Next, we'll implement the UI to support group blogging writing new UI unit tests along the way. We'll wrap up by documenting the new features.

### 4.1 Create UI unit testing infrastructure

Put UI unit testing framework in place and implement front end unit tests for at least the weblog edit, weblog categories, and website settings pages. We may do this using StrutsTest since we already have some tests based on Struts test.

- T: Put UI unit testing framework in place
- T: Implement UI unit tests for Weblog Entry CRUD
- T: Implement UI unit tests for Weblog Category CRUD

### 4.2 Backend changes

- T: Modify DDL create database and migration scripts
- T: WebsitePermissionsData, unit tests and implementation in UserManagerImpl
- T: ObjectAuditData, unit tests and AuditManagerImpl
- T: WebsiteData – inactive flag
- T: WeblogEntryData – new creator user\_id field, new status field
- T: Changes to UserManager
- T: Add auditing to WeblogEntry operations in WeblogManager

### 4.3 Pick up the webapp infrastructure pieces

Goal is to put Roller back together, no new UI – just make it work as it did before

- T: Changes and refactoring in RollerRequest object
- T: Changes to Struts Actions and JSPs: add `canSave()` checks to all actions
- T: Changes to PageModel, PageHelper and Velocity templates
- T: Ensure that all frontend unit tests and Roller itself runs as it did before

### 4.4 Implement group blogging UI

Implement the new actions, unit tests and pages

Development approach, for each page:

- Internationalization – all strings must be externalized using existing scheme
- Struts Dispatch Action – instead of many actions classes per page, uses a DispatchAction
- Page Model – instead of many request attributes, use a PageModel
- Action messages – Use action messages to convey success and error messages to user
- Prefer custom-tags over scriptlets – Use embedded Java code only as a last resort



- Prefer JSTL tags over Struts tags – Use JSTL tags instead of Struts tags wherever possible

Group blogging UI tasks:

- T: User Profile Action and JSP
- T: Website Permission Action and JSP
- T: Website Invite Action and JSP
- T: Website Creation Action and JSP
- T: Weblog Entry Editor Action and JSP changes
- T: Website Selection combo-box in Editor/Admin UI header
- T: Website-permission changes to custom Menu Tag
- T: Change CommentServlet to notify all Website members of new comment

#### **4.5 Other UI changes**

- T: Add checking for inactive websites in the main page actions, site wide newsfeeds, and in the planet subsystem.

#### **4.6 Side-by-side upgrade utility**

Allows for safe no-downtime upgrade from Roller 1.2 to Roller 2.0.

- T: Create side-by-side upgrade utility

#### **4.7 Testing**

- T: Load testing (include login, post, logout sequence this time – not just page/newsfeed hits)
- T: Acceptance testing (on staging server)

#### **4.8 Documentation**

- T: Create user documentation for each of the new pages