Managing User and Group Permissions

Recommended Role-Based Models

Table of Contents

1 Executive Summary

This document presents recommended role-based models for managing User and Group permissions in Graphite and suggests naming conventions for permission groups.

Graphite's permissions are defined on a per Project basis and allow fine-grained control of which concept Schemes and individual Properties each User Group can see and edit.

With this fine-grained control comes the need for Super Administrators and Project Administrators to define and manage User Groups tuned to the specific roles required by each Project. Managing User and Group permissions is a part of enterprise taxonomy governance and identity management policies in line with other enterprise systems integrated with the taxonomy management system.

Since each User Group in each Project contains a customizable set of permissions for each Scheme and set of Properties, it is also important that naming conventions are established for Groups in order to clearly convey what privileges are conferred upon Users when they become members of a Group.

This document defines archetypal roles and presents a baseline set of permissions associated with each role. It recommends three role-based models for managing User Group permissions.

The first and simplest model defines **generic roles** in which membership in a Group entitles Users with a set of permissions that are uniformly applied to all Schemes in a Project. The recommended naming convention for these Groups is [Project Name] + [Role Name].

The second model defines **Scheme-specific roles** in which membership in a Group entitles Users with a set of permissions that are specific to each Scheme. If it is required that some Users have view only access to one Scheme and editor access to another Scheme within the same Project, then this model can support the requirement by making those Users belong to more than one Group. The recommended naming convention for these Groups is [Scheme Name] + [Role Name].

The third model defines **multivalent roles** in which membership in a Group entitles users to different sets of permissions in relation to multiple Schemes. For example, membership in such a Group may confer view-only permissions to one Scheme and editor permissions to another Scheme. These multivalent role-based Groups are easy to configure. However, as the roles defined within a single Group become increasingly diverse it will become harder for Users to have a clear idea of what specific Scheme and Property permissions are conveyed by membership in such a Group. If multivalent role-based Groups are used then Super Administrators and Project Administrators should document each Group as part of the governance process, and determine naming conventions that best describe the purpose and scope of each Group.

2 Archetypal Roles

Each of the following archetypal roles are suggested and may change depending on the needs of your organization.

2.1 Super Administrator

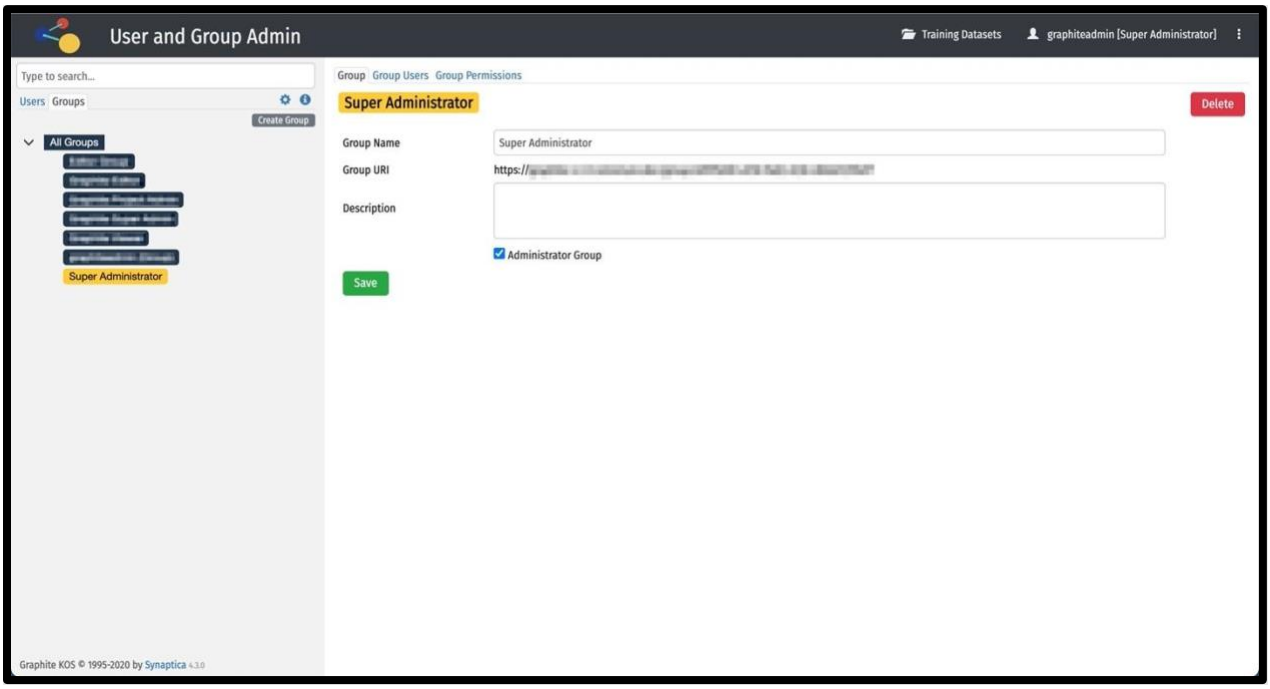
A Super Administrator (or System Administrator, Lead Taxonomist, or Technical Administrator) role includes all administrative functionality, including the ability to set up Namespaces, Templates, Schemes, Projects, and Users and Groups. Additionally, this role should have all taxonomy editorial functionality, including importing and exporting schemes, creating, editing, and deleting concepts, and all other tasks required to perform vocabulary work.

While there may be two different users, one business and one technical, requiring this role to be split out into a Taxonomy Super Administrator and a Technical Super Administrator, both roles would effectively have the same permissions.

The Super Administrator falls into the **generic roles** in which membership in a Group entitles Users with a set of permissions that are uniformly applied to all Schemes in a Project. Because the Super Administrator role crosses Projects, the name does not need to reflect Project names.

2.1.1 Typical Configuration Settings

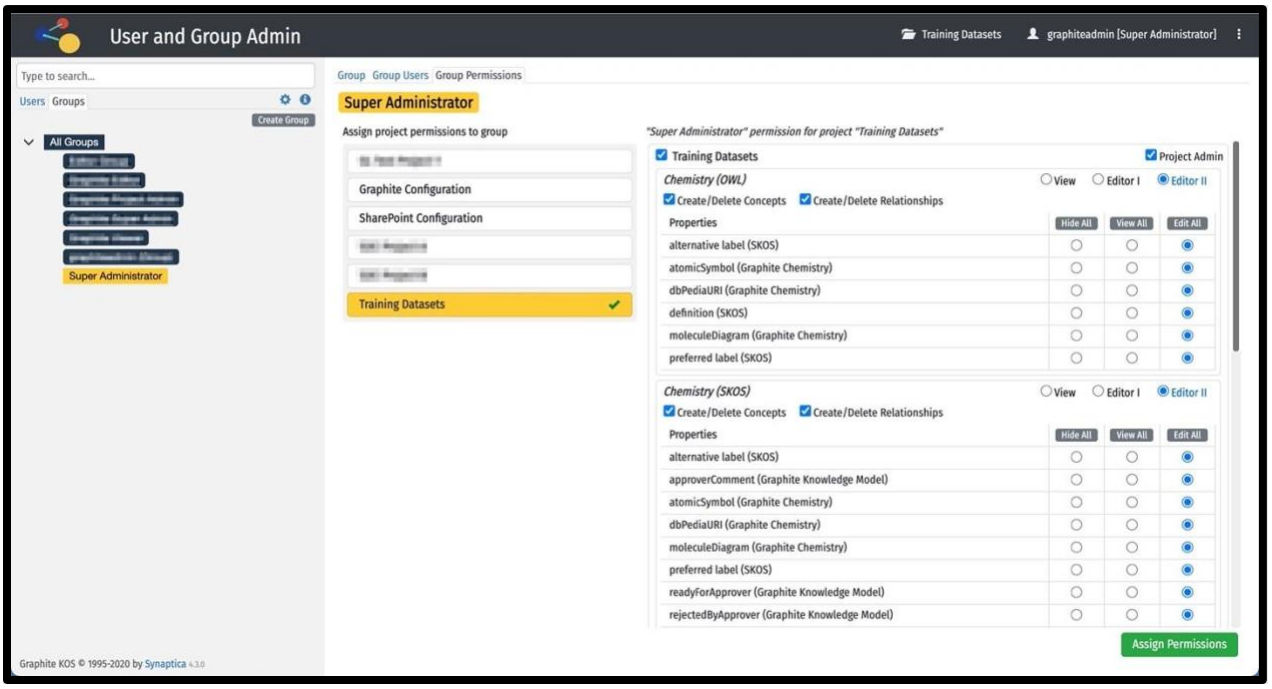
In the *Users and Groups Admin* menu, selecting the Super Administrator group with the *Group* tab open shows the group is set as an *Administrator Group*. User members of this group can be added in the *Group Users* tab.



In the *Group Permissions* tab, the users have permissions to the selected Project(s) and are a *Project Admin*. They also have *Create/Delete Concepts* and *Create/Delete Relationships, Editor II*, and *Edit All* permissions for all Schemes in the Project.

- Administrator Group: **Yes**/No
- Project: **Yes**/No
- Project Admin: **Yes**/No
- Create/Delete Concepts: **Yes**/No
- Create/Delete Relationships: **Yes**/No

- View: Yes/**No**
- Editor I: Yes/**No**
  - *Editor I* users can view and edit concepts but not change their approval Status.
- Editor II: **Yes**/No
  - *Editor II* users can view, edit and change the approval Status of concepts.
- Hide: Yes/**No**
- View: Yes/**No**
- Edit: **Yes**/No



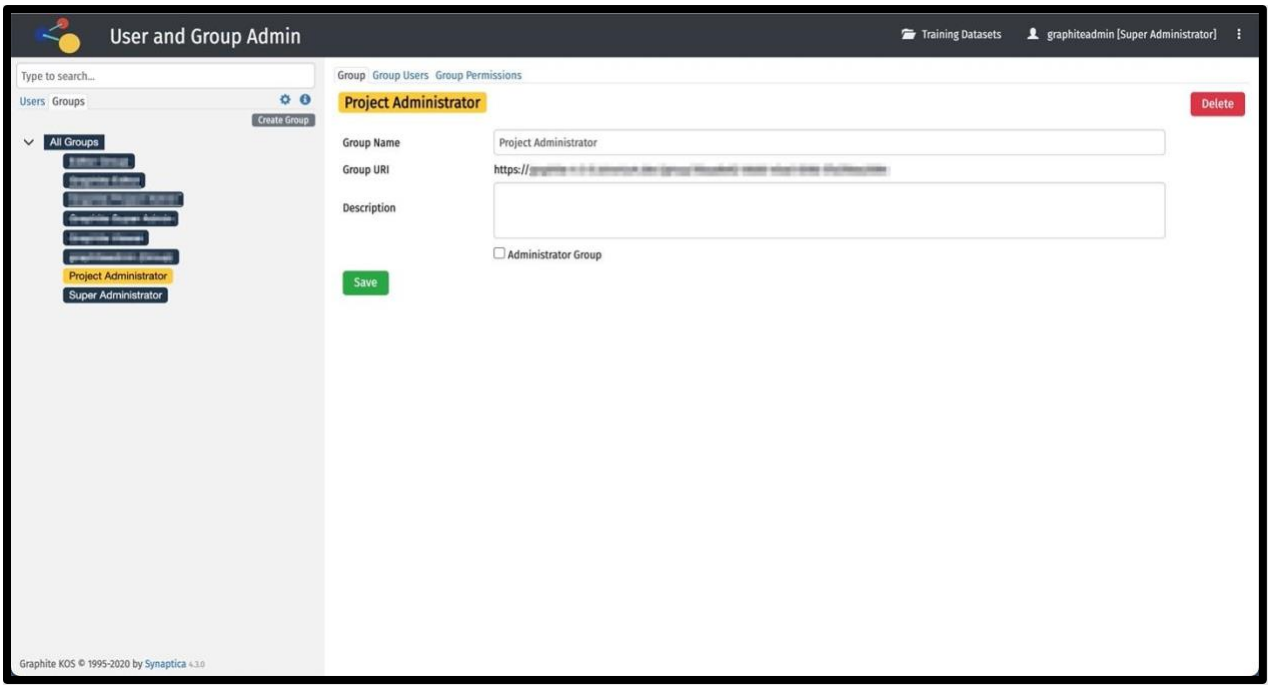
2.2 Project Administrator

A Project Administrator role includes the ability to manage one or more Projects and all taxonomy editorial functionality, including importing and exporting schemes, creating, editing, and deleting concepts, and all other tasks required to perform vocabulary work.

The Project Administrator falls into the **generic roles** in which membership in a Group entitles Users with a set of permissions that are uniformly applied to all Schemes in a Project. The recommended naming convention for these Groups is [Project Name] + [Role Name]. So, for example, *Training Datasets_Project Administrator*.

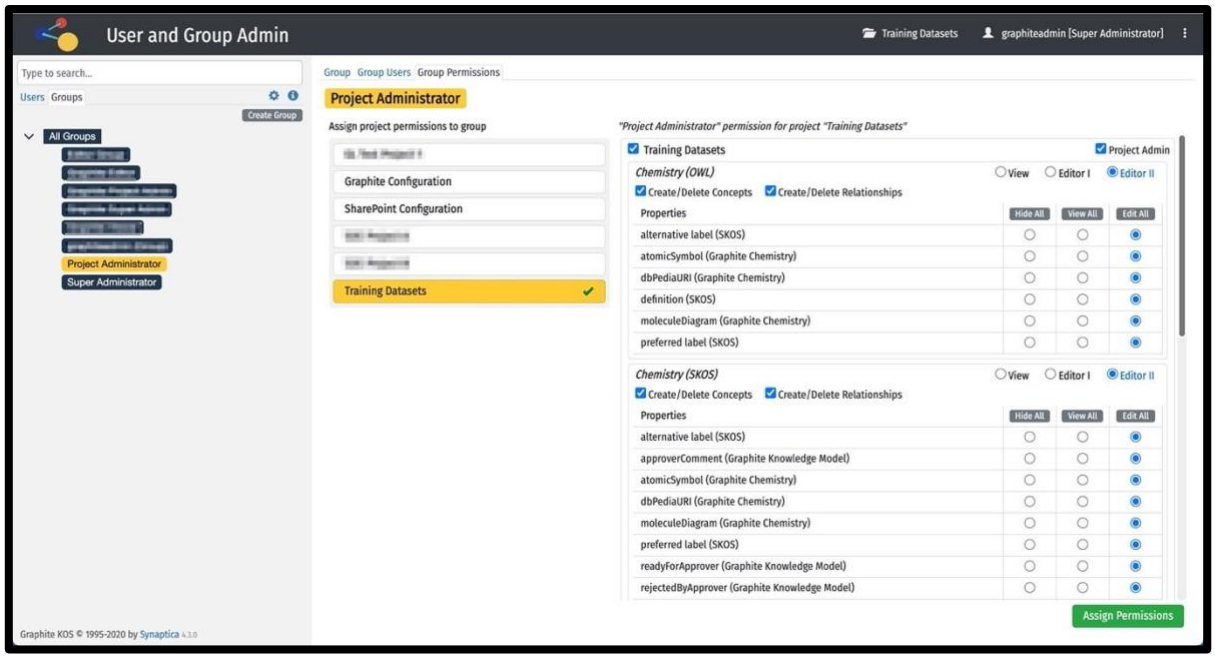
2.2.1 Typical Configuration Settings

In the *Users and Groups Admin* menu, selecting the Project Administrator group with the *Group* tab open shows the group is not set as an *Administrator Group*. User members of this group can be added in the *Group Users* tab.



In the *Group Permissions* tab, the users have permissions to the selected Project(s) and are a *Project Admin*. They also have *Create/Delete Concepts* and *Create/Delete Relationships, Editor II*, and *Edit All* permissions for all Schemes in the Project.

- Administrator Group: Yes/**No**
- Project: **Yes**/No
- Project Admin: **Yes**/No
- Create/Delete Concepts: **Yes**/No
- Create/Delete Relationships: **Yes**/No
- View: Yes/**No**
- Editor I: Yes/**No**
    - *Editor I* users can view and edit concepts but not change their approval Status.
- Editor II: **Yes**/No
    - *Editor II* users can view, edit and change the approval Status of concepts.
- Hide: Yes/**No**
- View: Yes/**No**
- Edit: **Yes**/No



2.3 Taxonomy Editor

A Taxonomy Editor (or Editorial Taxonomist) role includes all taxonomy editorial functionality, including importing and exporting schemes, creating, editing, and deleting concepts, and all other tasks required to perform vocabulary work. A Taxonomy Editor may be limited to a single Project or even particular Schemes within a single Project or across several Projects.

Within Graphite, there are two Editor settings, Editor I and Editor II. Both have the same permissions except an Editor I cannot change the status of a concept. The Taxonomy Editor could then be split into two roles: Taxonomy Editor I and Taxonomy Editor II with one set of permissions able to change the status of a concept as an approver and the other able to do all other work on concepts without the ability to change the status.

The Taxonomy Editor falls into the **generic roles** in which membership in a Group entitles Users with a set of permissions that are uniformly applied to all Schemes in a Project. The recommended naming convention for these Groups is [Project Name] + [Role Name]. So, for example, *Training Datasets_Taxonomy Editor I* or *Training Datasets_Taxonomy Editor II*.

2.3.1 Typical Configuration Settings

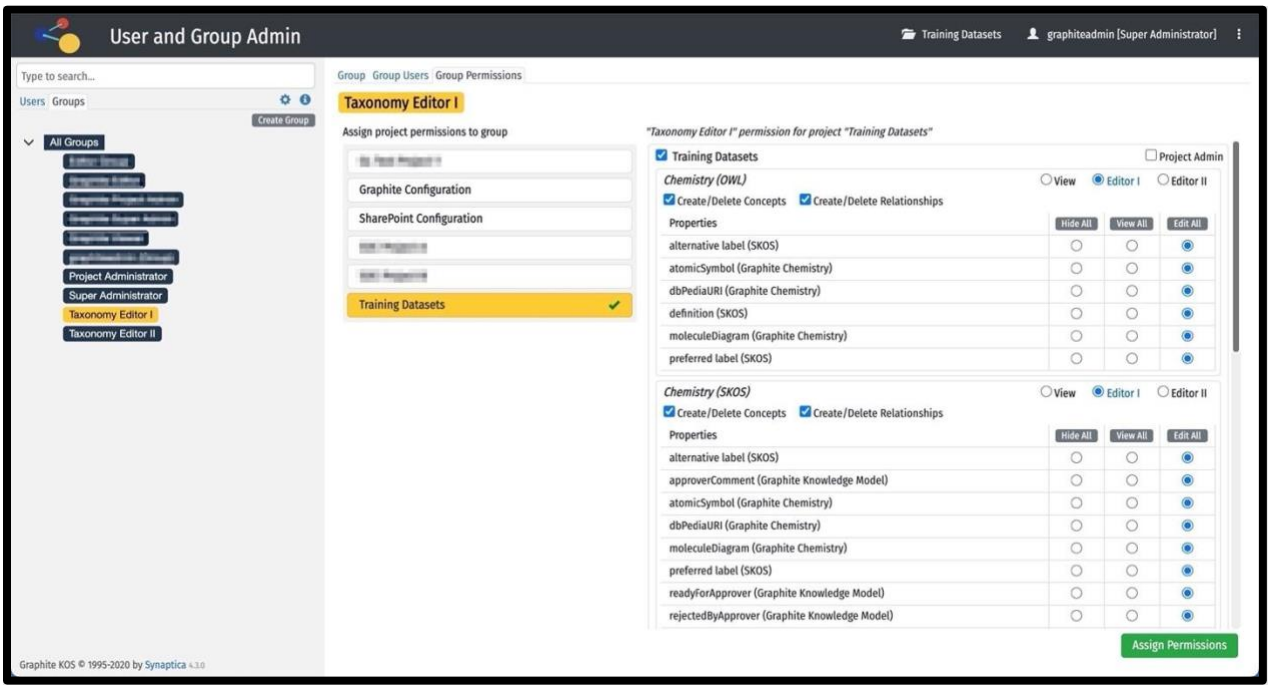
In the *Users and Groups Admin* menu, selecting the Taxonomy Editor I or Taxonomy Editor II group with the *Group* tab open shows the group is not set as an *Administrator Group*. User members of this group can be added in the *Group Users* tab.

In the *Group Permissions* tab, the users have permissions to the selected Project(s) but are not a *Project Admin.* They also have *Create/Delete Concepts* and *Create/Delete Relationships, Editor I* or *Editor II*, and *Edit All* permissions for all Schemes in the Project.

Permissions for a separate Taxonomy Editor I role are below.

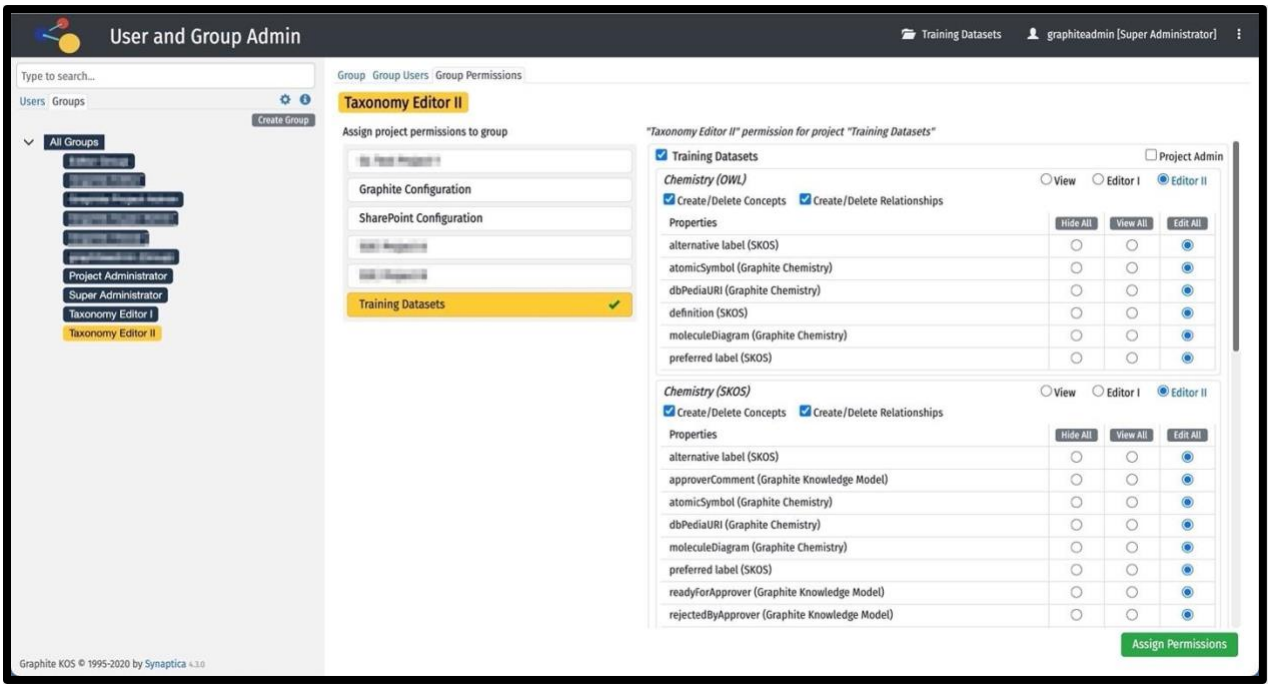
- Administrator Group: Yes/**No**
- Project: **Yes**/No

- Project Admin: Yes/**No**
- Create/Delete Concepts: **Yes**/No
- Create/Delete Relationships: **Yes**/No
- View: Yes/**No**
- Editor I: **Yes**/No
    - *Editor I* users can view and edit concepts but not change their approval Status.
- Editor II: Yes/**No**
    - *Editor II* users can view, edit and change the approval Status of concepts.
- Hide: Yes/**No**
- View: Yes/**No**
- Edit: **Yes**/No



Permissions for a separate Taxonomy Editor II role are below.

- Administrator Group: Yes/**No**
- Project: **Yes**/No
- Project Admin: Yes/**No**
- Create/Delete Concepts: **Yes**/No
- Create/Delete Relationships: **Yes**/No
- View: Yes/**No**
- Editor I: Yes/**No**
    - *Editor I* users can view and edit concepts but not change their approval Status.
- Editor II: **Yes**/No
    - *Editor II* users can view, edit and change the approval Status of concepts.
- Hide: Yes/**No**
- View: Yes/**No**
- Edit: **Yes**/No



2.4 Taxonomy Reviewer

A Taxonomy Reviewer role includes the ability to edit some properties and view others within a Scheme without the ability to add, delete, or change relationships for concepts. The Taxonomy Reviewer may include the ability to mark the concepts ready for review.

The Taxonomy Reviewer falls into the **Scheme-specific roles** in which membership in a Group entitles Users with a set of permissions that are specific to each Scheme. If it is required that some Users have view only access to one Scheme and editor access to another Scheme within the same Project, then this model can support the requirement by making those Users belong to more than one Group. The recommended naming convention for these Groups is [Scheme Name] + [Role Name]. So, for example, *Chemistry (OWL)_Taxonomy Reviewer* and *GEMET Thesaurus_Taxonomy Reviewer* in which each group has the same *Editor I* setting, but specific *Hide*, *View*, and *Edit* settings differ per property.

The Taxonomy Reviewer also falls into the **multivalent roles** in which membership in a Group entitles users to different sets of permissions in relation to multiple Schemes. For example, membership in such a Group may confer view-only permissions to one Scheme and editor permissions to another Scheme. While infrequent, it is possible a Taxonomy Reviewer may have one set of permissions for a Scheme in one Project and another set of permissions for the same Scheme in another Project based on organizational requirements. In such cases, the organization should determine how to differentiate the Taxonomy Reviewer per Scheme per Project.

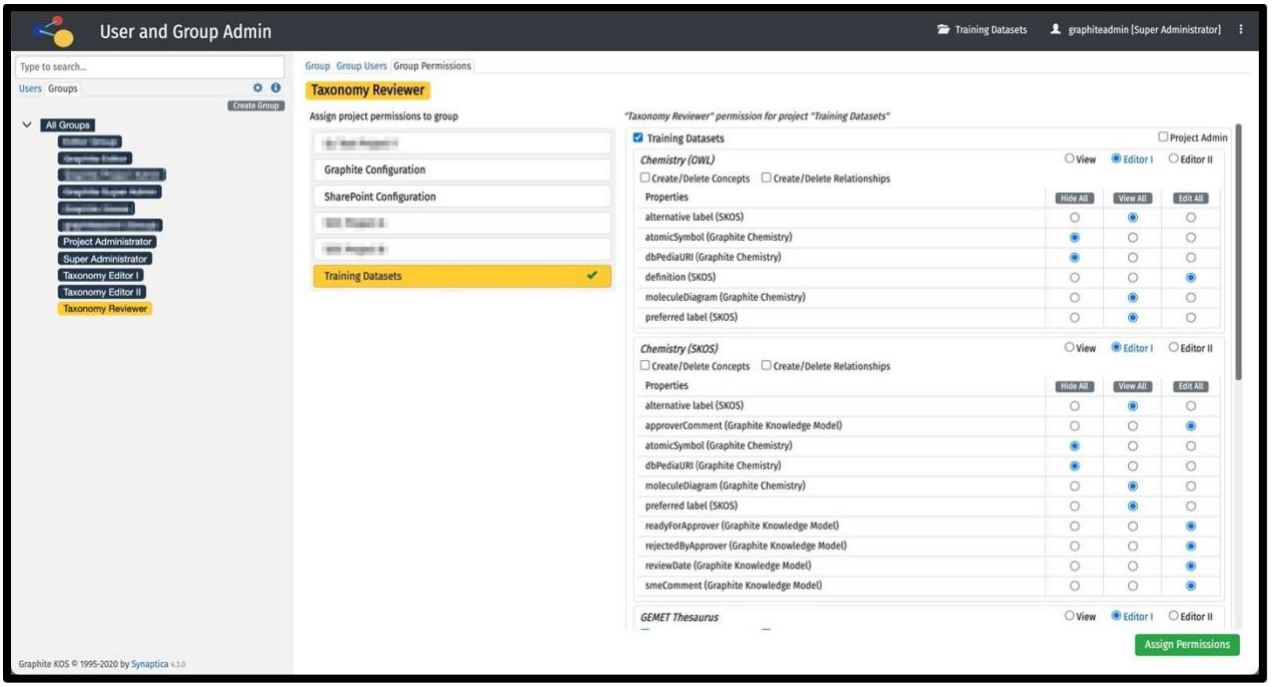
2.4.1 Typical Configuration Settings

In the *Users and Groups Admin* menu, selecting the Taxonomy Reviewer group with the *Group* tab open shows the group is not set as an *Administrator Group*. User members of this group can be added in the *Group Users* tab.

In the *Group Permissions* tab, the users have permissions to the selected Project(s) but are not a *Project Admin*. They do not have *Create/Delete Concepts* and *Create/Delete Relationships*. This role is *Editor I*, and have a mix of *Hide*, *View*, and *Edit* permissions for selected Schemes in the Project as required.

Below is an example of granular property permissions a Taxonomy Reviewer may have.

- Administrator Group: Yes/**No**
- Project: **Yes**/No
- Project Admin: Yes/**No**
- Create/Delete Concepts: Yes/**No**
- Create/Delete Relationships: Yes/**No**
- View: Yes/**No**
- Editor I: **Yes**/No
    - *Editor I* users can view and edit concepts but not change their approval Status.
- Editor II: Yes/**No**
    - *Editor II* users can view, edit and change the approval Status of concepts.
- Hide: **Yes**/No
- View: **Yes**/No
- Edit: **Yes**/No



2.5 Taxonomy Viewer

A Taxonomy Viewer role includes the ability to view certain Schemes, subsets of concepts within Schemes, or certain properties or relationships within concepts.

The Taxonomy Viewer falls into the **Scheme-specific roles** in which membership in a Group entitles Users with a set of permissions that are specific to each Scheme. The recommended naming convention for these Groups is [Scheme Name] + [Role Name]. So, for example, *Chemistry*

(OWL)_Taxonomy Viewer and *GEMET Thesaurus_Taxonomy Viewer* in which each group has the same *View* setting, but specific *Hide* and *View* settings differ per property.

The Taxonomy Viewer also falls into the **multivalent roles** in which membership in a Group entitles users to different sets of permissions in relation to multiple Schemes. For example, membership in such a Group may confer view-only permissions to one Scheme and editor permissions to another Scheme. While infrequent, it is possible a Taxonomy Viewer may have one set of permissions for a Scheme in one Project and another set of permissions for the same Scheme in another Project based on organizational requirements. In such cases, the organization should determine how to differentiate the Taxonomy Viewer per Scheme per Project.

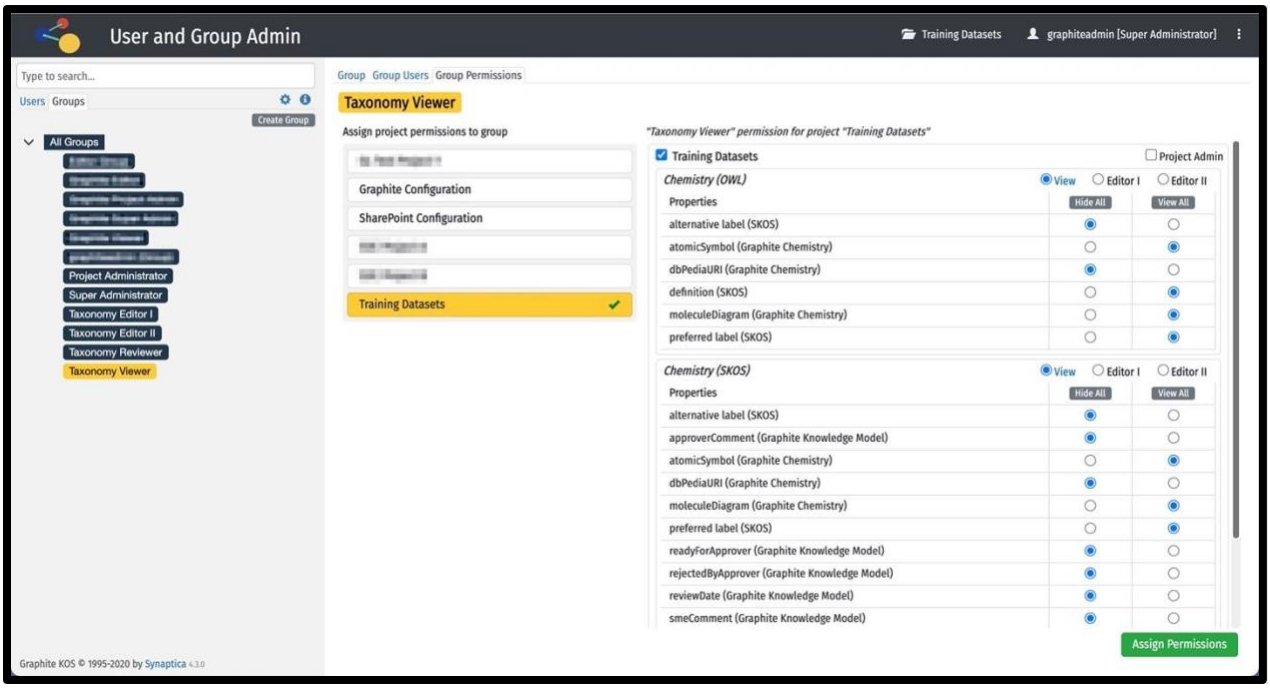
2.5.1 Typical Configuration Settings

In the *Users and Groups Admin* menu, selecting the Taxonomy Viewer group with the *Group* tab open shows the group is not set as an *Administrator Group*. User members of this group can be added in the *Group Users* tab.

In the *Group Permissions* tab, the users have permissions to the selected Project(s) but are not a *Project Admin*. They do not have *Create/Delete Concepts* and *Create/Delete Relationships*. This role is *View* and have a mix of *Hide* and *View* permissions for selected Schemes in the Project as required.

Below is an example of granular property permissions a Taxonomy Viewer may have.

- Administrator Group: Yes/**No**
- Project: **Yes**/No
- Project Admin: Yes/**No**
- Create/Delete Concepts: Yes/**No**
- Create/Delete Relationships: Yes/**No**
- View: **Yes**/No
- Editor I: Yes/**No**
    - *Editor I* users can view and edit concepts but not change their approval Status.
- Editor II: Yes/**No**
    - *Editor II* users can view, edit and change the approval Status of concepts.
- Hide: **Yes**/No
- View: **Yes**/No
- Edit: Yes/**No**



3 Workflow for Managing Groups for Generic Roles

From the generic roles, the Super Administrator(s) and Project Administrator(s) should be configured before all other groups. A Super Administrator is required to access the system and make configurations in order to allow other users access to Projects.

Once complete, Project Administrators can begin work configuring Schemes within Projects for other roles to perform their work. Project Administrators can assign permissions to users in Taxonomy Editor, Taxonomy Reviewer, and Taxonomy Viewer groups.

4 Workflow for Managing Groups for Scheme-Specific Roles

Scheme-specific roles are configured by Super Administrators and Project Administrators, though it is more likely that the latter will assign permissions to Taxonomy Editor, Taxonomy Reviewer, and Taxonomy Viewer groups.

5 Workflow for Managing Groups for Multivalent Roles

Multivalent roles are the most organization-specific and challenging roles to manage as both group names and permissions may vary from Scheme to Scheme and Project to Project.

User and Group names must be unique, so determining a naming scheme which fits the use case in the organization must be made part of the taxonomy governance process. In addition, since users rarely have access to only Graphite, especially when Single Sign-On (SSO) is configured,

these user names must be consistent with usage in other systems, whether this be by name, email address, or other user identifier.

In addition to the archetypal roles suggested in this document, multivalent and organizationally-specific user groups could be created with varying permissions per Scheme and Project. For these roles, there are several options which may help in consistent and clear Group naming policies.

- By organization within the enterprise. These may be the names of teams, organizational units or verticals, or other internal organizing principals. For instance, *Marketing Taxonomy Reviewer* or *Finance Project Administrator*.
- By organization project. Since taxonomy projects are often launched in conjunction with other information and knowledge management projects, group names could be aligned with these projects. Bear in mind, however, that projects are time sensitive and may require changes when projects begin or end. For example, *KM Platform Launch Taxonomy Reviewer* or *Text Analytics Taxonomy Editor*.
- By Graphite Project. Adding the name of the Graphite Project in addition to the role is also possible. Users are cautioned against creating overly long or cumbersome names for the sake of usability. For example, *Marketing Project_Taxonomy Editor I* may be descriptive enough without adding the name of Schemes within the Project.
- By system integration. Since Graphite is meant to be a central ontology management system integrated with consuming systems, there may be system-specific roles. If users are more accustomed to thinking about where taxonomies are surfaced rather than the content of the vocabularies, system-based Group names may be preferable. For example, *SharePoint_Project Administrator* or *Confluence_Taxonomy Editor II*.

For all of the above suggestions, Group names should generally be kept as simple and as concise as possible. While Graphite Group name text fields can support many characters, usability becomes a factor in overly long or descriptive Group names.

As multivalent roles become more numerous and more specific, governance policies will be essential to managing naming conventions and use cases. Even then, assigning the same user to conflicting permissions will be possible. In such cases, the highest permission settings are applied and for each user the Scheme and property permissions will be displayed in red if conflicting permissions are granted.