Permissions Best Practice Guide

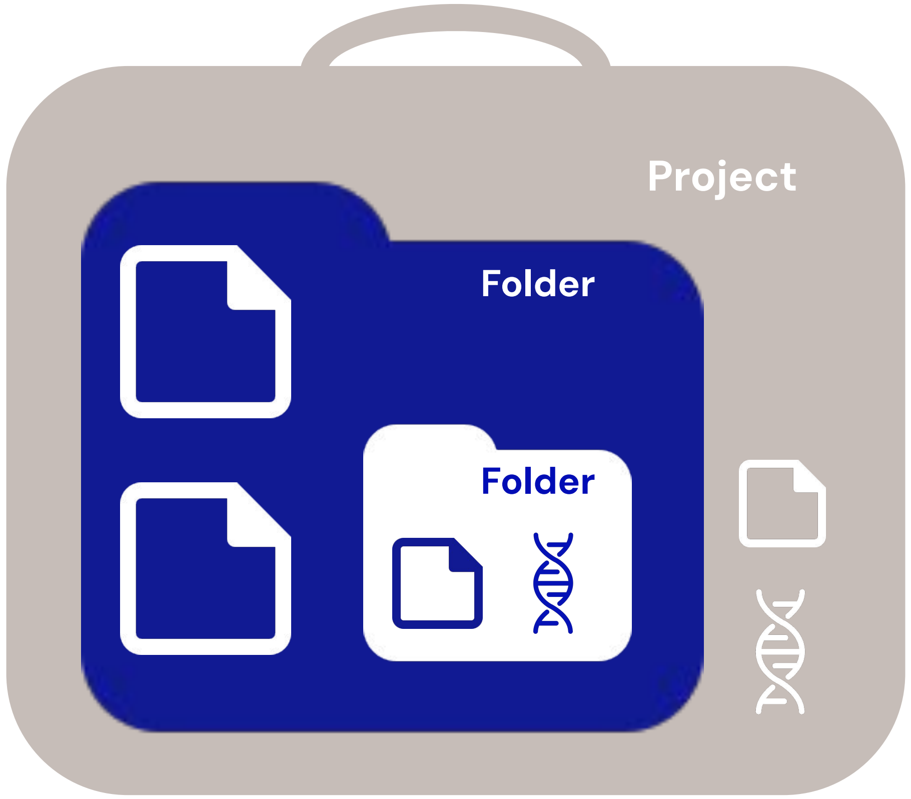
Benchling

Understanding Permissions in Projects, the Registry, and Inventory

1. **Factors Influencing Permissions:**
   a. Object Type
   b. Registration Status
   c. Schema Settings

2. Objects may be present in multiple locations but derive permissions from one location

3. Permissions are based on the current state of the object, regardless of its history



How should you define & manage your Permissions Structure?

Which Permissions Structure is Best Suited for You?

All Permissions structures have trade-offs so ask yourself what is most important

Gather your Benchling admins/leads and ask yourselves the following questions:

1. Do we have a defined team structure?
2. What is the level of collaboration needed at our company?
3. How do we want to structure our Projects and Registry?
4. What permissions do we want our entities to follow - Project based? Registry Based?
5. How granular do we need our permissions to be?

Effectively managing your users and organization is vital to maintaining a good permissions maintenance; users should be able to access/upload the correct data when needed:

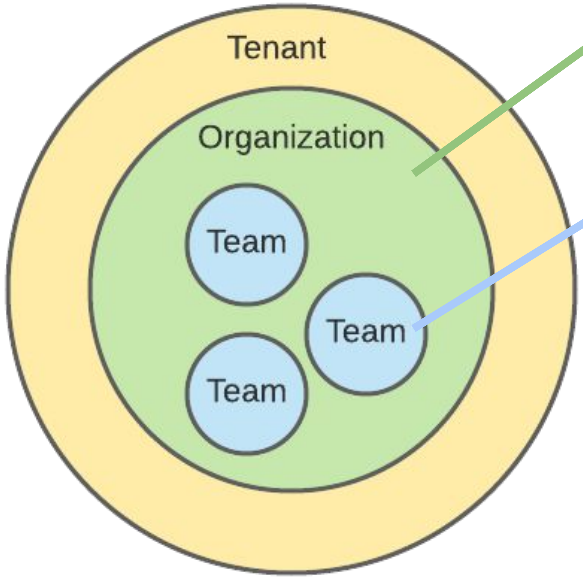
The Tenant Admin Console (TAC) is only accessible by Tenant Admins. Here you can grant specific users access to the entire tenant and manage your users, teams and organizations

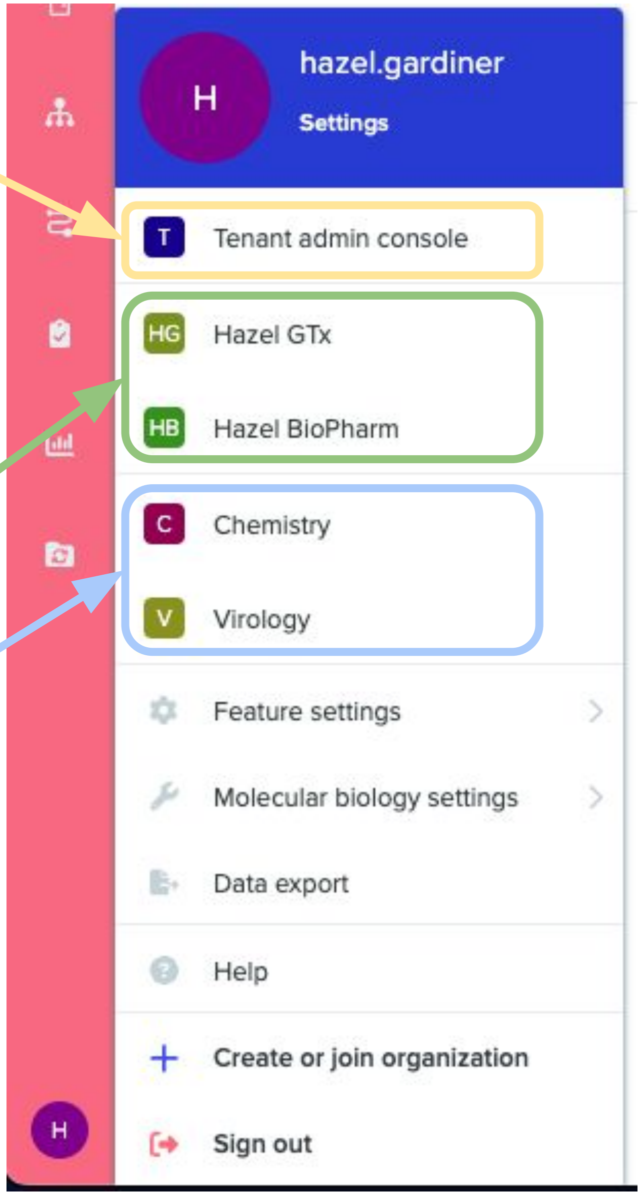
Organizations and Teams

Your entire lab or company can be represented in Benchling as an organization

Teams provide an additional level of user management. These teams can be collectively granted specific permissions as a group

- We recommend that teams are aligned by one of the following:
  - Function
  - Target
  - Program




hazel.gardiner
Settings

Project vs Registry Level Permissions

Project & Folder*-level Permissions in Benchling

	Read	Append	Write	Admin
Edit Project/Folder Permissions	🚫	🚫	🚫	✅
Create Folders (within Projects)	🚫	✅	🚫	✅
Add/remove Auditors	🚫	🚫	🚫	✅
View Entries	✅	✅	✅	✅
See Version History & Timestamps	✅	✅	✅	✅
Create New Entries & Data	🚫	✅	✅	✅
Edit Entries & Data	🚫	🚫	✅ if author	✅
Archive Entries & Data	🚫	🚫	✅ if author	✅

* All users with Admin permission can edit Entries for a particular Project regardless if they are an Entry Author

* As of 05/06/2024, this feature is currently in limited availability. If you don't have access to manage folder collaborators, contact Customer Support to request access.

Registry-level Permissions in Benchling

	Read	Append	Write	Admin
View registered entities	✅	✅	✅	✅
Register new entities	🚫	✅	✅	✅
Edit register entities	🚫	🚫	✅	✅
Unregister entities	🚫	🚫	✅	✅
Create / Edit Schemas & Change Registry Permissions	🚫	🚫	🚫	✅

If there are overlapping permissions or access policies assigned to users, teams, organizations or registries, Benchling allows the **most permissive** access policy to that user or team.

Benefits & Trade-offs of Schema Permissions

Should you share schemas and fields across your entire organization?

	Org Shared Schemas	**Team-Specific Schemas**
Benefits	+ Lower admin lift to develop and maintain + Less confusion for users with fewer schemas to Choose + Cleaner UI Experience	+ Schemas designed to fit to one team's needs + Filter by schema to view team-specific metadata + Can utilize schema-specific permissions + Descriptive schema names
Trade-offs	- Team Specific fields are shown to everyone - Filtering is more complicated to identify specific teams' entities - Aggregated datasets may have missing metadata values	- Increase in number of schemas increases Admin Lift to maintain - Long list of schemas to choose from for users - Downstream entities may need to be developed for each team's outputs, increasing complexity - Possible redundancy or entity duplication if different teams are working with and registering the same samples

Note: you can check if your schema is using Registry or Project based permissions by navigating to the schema in Feature Settings and viewing the Access Policies tab. If you'd like to change this setting on a schema, please reach out to your account team or Benchling Support.

Permissions Best Practices

Best Practices: Permissions Structure

Focus	Why?
Clear Role Definitions	Define roles such as Admin, Editor, Viewer, etc., with distinct permissions based on job responsibilities/team structure.
Granular Permission Settings	Utilize granular permissions at the project, folder, or item level to control access. Consider factors such as object type, registration status, and schema settings. For example, in addition to project-level permissions, Benchling offers the capability to set permissions at the folder level. Admins can leverage folder-level permissions to grant users access to specific folders without providing access to the entire project. These permissions stack onto project permissions, allowing for fine-grained control over access rights. By applying the least permissive access policies to projects and granting additional access to their contained folders as needed, organizations can ensure data security while facilitating efficient collaboration.
Regular Review and Updates	Conduct regular reviews to align permissions with organizational changes and project requirements.
Address Conflicting Policies	Be aware of conflicting policies as users are granted the most permissive access when multiple policies apply. Resolving conflicting policies ensures consistent access control and prevents unintended permission on user actions within Benchling.

Resources for Permissions

Permissions for Objects in Projects, Registries, and Inventory

Benchling Support

Benchling Learning Labs - User Management, Access, and Permissions

Benchling Community