Security Guide

Revision Date
November 22, 2022

This guide has consolidated security-related information from several guides, both at the user and administrator levels. Information pertinent to administrators will be indicated as such.

Author
Martin Reisinger

Comments/Suggestions:
Please contact Martin Reisinger, either via feedback@ars-grin.gov, marty.reisinger@usda.gov or mar@rrginc.com with any suggestions or questions related to this document. This and other GRIN-Global – related documentation can be downloaded from the GRIN-Global Project website Documentation Page: https://www.grin-global.org/userdocs.htm

The Appendix contains this document's revision notes.

Table of Contents

Security: Ownership & Permissions

Organizations typically have very unique security needs; GRIN-Global (GG) was designed to be flexible enough to accommodate these diverse needs. In GG, when speaking of security, there are two primary concepts that intersect:

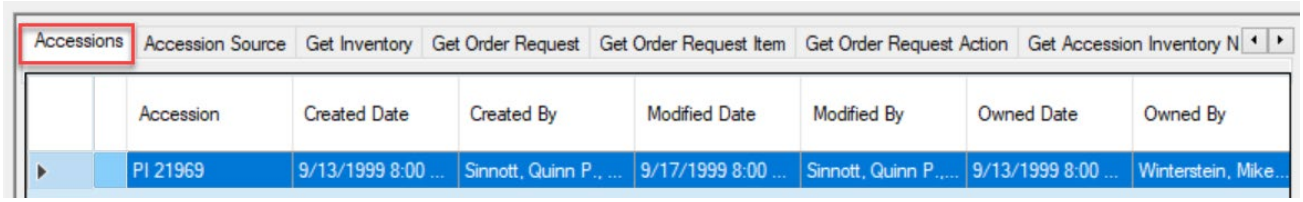
- ownership
- permissions

Ownership

An owner *usually* can update or delete records which she has created. Several points to remember:

- only one owner per record
- an owner can transfer ownership to another user
- an owner can provide permissions (Read, Update, Delete) to multiple users

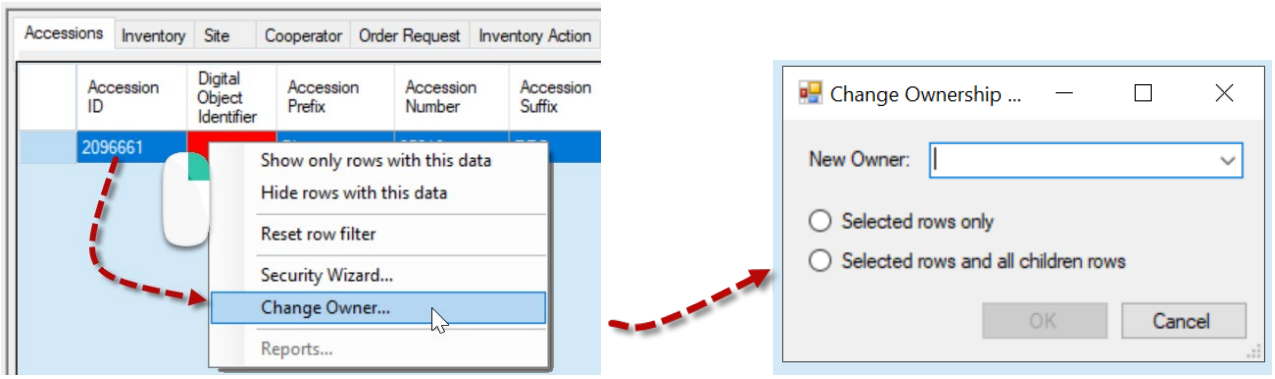
In the Curator Tool, in most dataviews, the six columns displayed at the far right are the "audit fields" that inidicate who created the record, who modified it (if it was), and who owns it. Three of the six fields are the associated dates fields.



In some cases, the person creating the record is not necessarily the owner of the record. For example, **Inventory** records are assigned the same owner as the owner of the **Inventory Maintenance Policy** that was used to create the Inventory record.

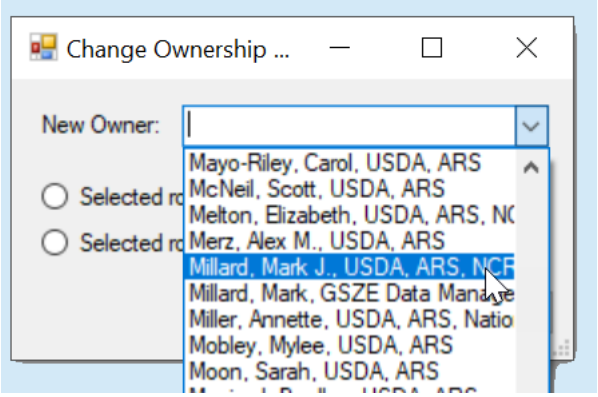
Changing Ownership

Changing ownership in the Curator Tool is simple; if you own the record, select it; right-click:



Typically, you would select "**Selected rows and all children rows**."

You will be prompted to select the new owner from a list of the organization's internal users:



SQL for Determining Ownership

Under Tools on the Public Website, (when logged in), you can run a query to determine ownership hierarchy (For more details, see the Administrator section.)

```
SELECT st1.table_name AS child, st2.table_name AS owner
FROM sys_table_relationship str
JOIN sys_table_field stf1 ON stf1.sys_table_field_id =  str.sys_table_field_id
JOIN sys_table st1 ON st1.sys_table_id = stf1.sys_table_id
JOIN sys_table_field stf2 ON stf2.sys_table_field_id = str.other_table_field_id
JOIN sys_table st2 ON st2.sys_table_id = stf2.sys_table_id
WHERE relationship_type_tag = 'OWNER_PARENT'
```

Changing Permissions

If you own the record, you can grant specific permissions to designated users so that they may edit the record, and even delete the record.

A permission restricts or grants access to a resource in GRIN-Global; for a Curator Tool user, a resource is typically a row in a table – a record displayed within a dataview.

You use the security wizard to establish permission levels to protect specific record types from accidental (or intentional) deletion. For example, you can establish security permissions so that specific staff in the organization will be able to update specific inventory records, but not delete them.

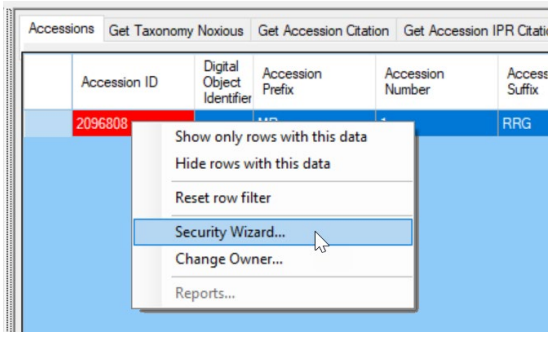
In the **Ownership** section above, we discussed how the creator of a record doesn't necessarily own the record. Inventory records are owned by the owner of the **Maintenance Policy** that was used when creating the **Inventory** record. Therefore, someone creating an Inventory record might not have permission to update or delete it.

In another example, a genebank may have student technicians uploading (creating) observation information or creating action records on an accession or inventory, but the genebank does not want the student to alter the data. A permission policy can be established to ensure that this environment is set up to safeguard the data.

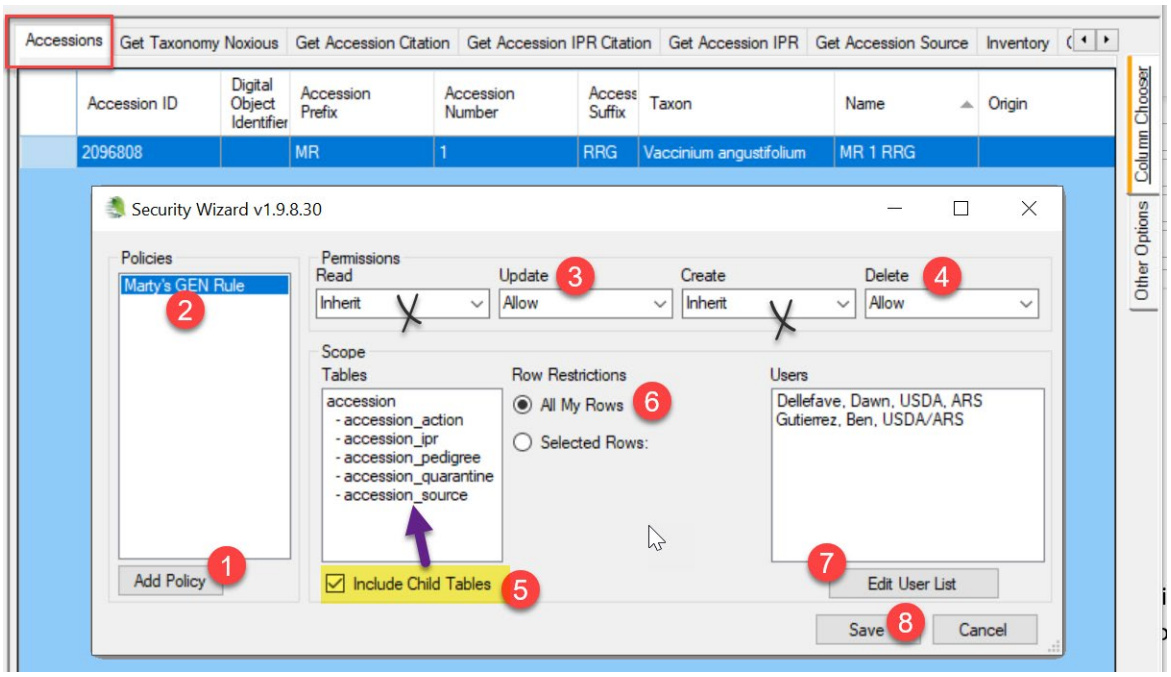
Steps to Establish Permissions

1. Determine what records you wish to protect: accessions, inventory, orders, etc. Open the respective dataview.

2. In the Curator Tool, locate a record (or records) you own – then right click – select **Security Wizard**



3. In the **Security Wizard**, start from left to right. (image next page)
    a. (1) Add a Policy – (2) Rename the policy (optional) (right click).
    b. (3-4) Change **Update** and **Delete** to ***Allow*** (ignore **Read** and **Create** does not need to be changed, ever)
    c. (5) Include all children tables
    d. (6) All rows
    e. (7) Select who is to receive the permissions
    f. (8) Save



This **Security Wizard** screen has two fields (crossed over in the image above) which typically do not apply to the Curator Tool user. "Read": Most organizations allow the internal staff to read all records. "Create" doesn't apply either, because in the context of selecting records to assign permissions, the records already exist.

Note

Currently the inheritance only cascades one level. Under Scope, you see the child tables to Accession. This implies that it may be necessary for you to establish certain permissions at the accession level, and then again at the inventory level. As needed, you should consider repeating steps 2 for Inventory, and then for Orders, then for Observations.

Tip

In the **Row Restrictions** option (labeled #6), select **All My Rows** to guarantee that records created in the future will also be governed by this policy.

Permission Definitions

The permission definitions are defined in the tables below. To simplify all of this, remember that most permission situations involve allowing or denying users to do certain things – reading, deleting, or updating records. Example: you may want certain users to be able to update "my" inventory records, but never delete them.

Permission Defined

A permission restricts or grants access to a resource in GRIN-Global. A resource is defined as a specific table, dataview, or row. A permission defines four kinds of rights:

A permission of type:	Has the ability to:
Read	Read existing data
Update	Update existing data
Delete	Delete existing data
Create*	Insert *new* data

* in the CT, ignore this option – it really doesn't apply; typically, you will set the Update and Delete options since usually within an organization everyone internally should be able to read the records

Each permission can have one of three values:

Value	Description
Allow	Allows access
Deny	Denies access
Inherit	Neither allows nor denies access; access is situational; it is inherited from a previous definition (typically the permission value of the parent table)

SQL for Determining Permissions

Under the Public Website Tools option, you can determine what permission policies have been created for a user.

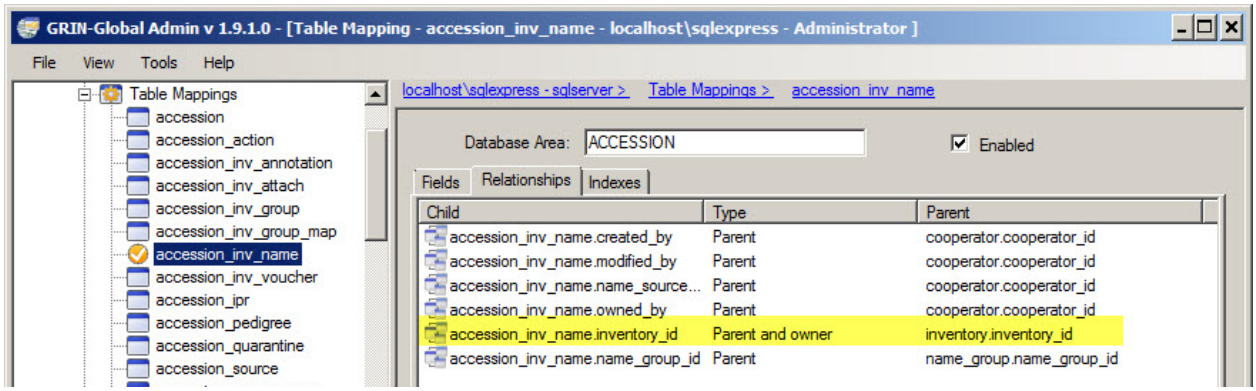
The "SW" in the example above is indicating that the permission was created via the Curator Tool's Security Wizard.

```
SELECT su.user_name, sg.group_tag, sp.permission_tag, st.table_name, CONCAT(c.first_name, ' ',
c.last_name) AS owner
FROM sys_user su
JOIN sys_group_user_map sgum ON sgum.sys_user_id = su.sys_user_id
JOIN sys_group sg ON sg.sys_group_id = sgum.sys_group_id
JOIN sys_group_permission_map sgpm ON sgpm.sys_group_id = sg.sys_group_id
JOIN sys_permission sp ON sp.sys_permission_id = sgpm.sys_permission_id
LEFT JOIN sys_table st ON st.sys_table_id = sp.sys_table_id
JOIN cooperator c ON c.cooperator_id = sp.created_by
WHERE sp.owned_by != 48
 AND user_name LIKE '%reisinger%'
```

This Section is Primarily Relevant to Administrators

Parent and Owner Relationships Between Dataviews

In the **Admin Tool**, relationships are mapped between dataviews. For instance, there is a relationship from accession to accession_inventory_name with the Relationship Type defined as "Parent and owner."



When relationships are mapped between dataviews, the children tables inherit the security settings of the parent. This means if someone creates a record in **accession_inventory_name**, the owner is the same as the owner of the parent record, in this case the **inventory** record.

When no relationship of "Parent and owner" has been defined, then the creator is the owner. When doing ownership calculation, relationships *are* taken into consideration.

The following question is often asked: “What is the rationale behind having the Inventory Site derived from whoever owns the inventory maintenance policy?” The GG designers felt it worked better to assign inventory ownership based on the Maintenance Policy that was used in making the Inventory record than on the Accession to which the inventory was linked. Inventory records are assigned the same owner as the owner of the **Inventory Maintenance Policy** used to create the Inventory record. This enables another site, such as a backup site that manages inventory for accessions owned elsewhere, to be able to use their own **Inventory**

Maintenance Policy to assure they can then maintain any inventory records that they create. They own the inventory, but the accession is owned elsewhere.

Background on the Parent Method of Assigning Ownership

The basic ownership model is that anyone can create a record in any table. The creator of the record is then typically the owner – that person is the only person who can modify or update the record (except the system administrator). GG was designed so that some subsidiary tables would have ownership that flowed from a parent table, so that the owner of the important parent record would be able to manage the subsidiary child tables. This is controlled by an OWNER-PARENT setting in the **sys_table_relationship** table that indicates which table, if any, acts as the owning parent.

It is helpful for record management to have all the rows in the associated accession tables owned by the same person and the same is true for the inventory area. Originally, GG was going to have the inventory obtain its ownership from accession, but that caused issues with backup site inventory records, so inventory ownership was switched to the maintenance policy.

If that is an issue for a genebank, a change could be made to the **sys_table_relationship** table to use accession as the owner of inventory or simply drop setting ownership by parent for the inventory table. That setting can be adjusted with the Admin Tool. In the AT, go into Table Mappings for the **Inventory** table and use the **Relationships** tab which shows all the Foreign Key links. Look for the "**Parent and owner**" Type and change it to "**Parent**."

There may be tables that are currently stand-alone that could benefit from adding a Parent and owner relationship. The following SQL shows the tables that currently get ownership from a parent table:

```
SELECT st1.table_name AS child, st2.table_name AS owner
FROM sys_table_relationship str
JOIN sys_table_field stf1 ON stf1.sys_table_field_id = str.sys_table_field_id
JOIN sys_table st1 ON st1.sys_table_id = stf1.sys_table_id
JOIN sys_table_field stf2 ON stf2.sys_table_field_id = str.other_table_field_id
JOIN sys_table st2 ON st2.sys_table_id = stf2.sys_table_id
WHERE relationship_type_tag = 'OWNER_PARENT'
```

child	owner
accession_action	accession
accession_inv_annotation	inventory
accession_ipr	accession
accession_pedigree	accession
accession_quarantine	accession
accession_source	accession
accession_source_map	accession_source
accession_inv_voucher	inventory
crop_trait_code	crop_trait
crop_trait_observation	inventory
inventory	inventory_maint_policy
inventory_action	inventory
inventory_quality_status	inventory
inventory_viability	inventory
order_request_action	order_request
order_request_item	order_request
taxonomy_genus	taxonomy_family
genetic_marker	crop
genetic_observation	inventory
accession_inv_attach	inventory
accession_inv_name	inventory
order_request_attach	order_request
accession_inv_group_map	accession_inv_group
geneva_site_inventory	inventory
nc7_site_inventory	inventory
ne9_site_inventory	inventory
nssl_site_inventory	inventory
opgc_site_inventory	inventory

child	owner
parl_site_inventory	inventory
s9_site_inventory	inventory
w6_site_inventory	inventory
accession_inv_group_attach	accession_inv_group
method_attach	method

Adding Users and Implementing Security (for Administrators)

Purpose

This section contains detailed steps for including information on setting security permissions, as well as a summary for adding new users to GRIN-Global. It summarizes the basic considerations needed when establishing new UserIDs, as well as general security considerations.

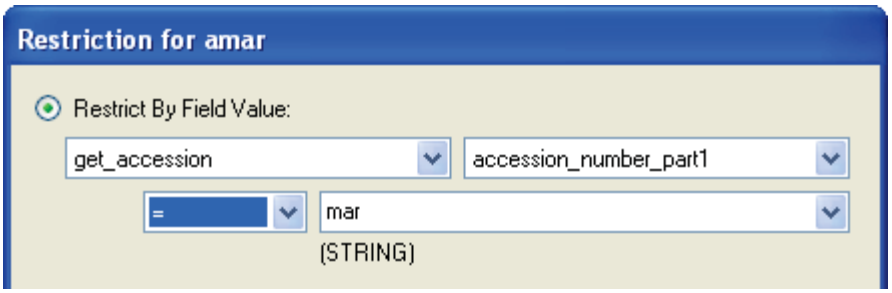
Overview

In general terms, there are three security alternatives to be considered:

1. Disable security entirely
2. Have an intermediate level of security centered around parent and child tables or related dataviews
3. Very strict security, as controlled as possible, where security is set at the record level based on specified criteria

The first scenario is discussed on the next page. You can accomplish the second alternative by setting up permission groups as needed and including Users in these groups or by establishing specific permissions and then applying them as needed to individual UserIDs; this is also explained on the next page. The third case, which typically controls a user's access to particular records, may involve **Permissions** based on criteria that evaluate a field's contents.

For example, the permission being established in this **Restriction** screen is based on the contents in the **accession_number_part1** field. The user being granted this permission would be limited to accession records whose **accession _number_part1** field was equal to "mar."



This **Restriction** screen is available in the Admin Tool when adding new permissions. (The Curator Tool users who are records owners also have similar screens available via the CT Security Wizard. A record owner can indicate whether other users can read, update (edit), write, or delete data at the record level.)

Disable Security Entirely

One (fairly drastic) option is to disable security entirely:

Action	Description
Disable or enable security *system wide*	Only the administrators of the installation can disable/enable security. In the Admin Tool, on the **Web Application** node, right-click **DisableSecurity**; select **Properties**; set **DisableSecurity** to **true**. This action disables security across everything and all permissions are ignored.

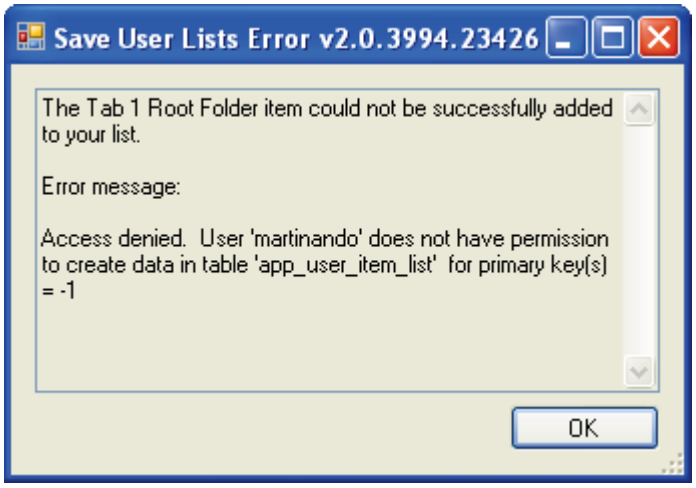
Adding New GG Users

Refer to the Admin Tool Guide for complete "how-to" directions on setting up new users. Some key points to remember when adding users:

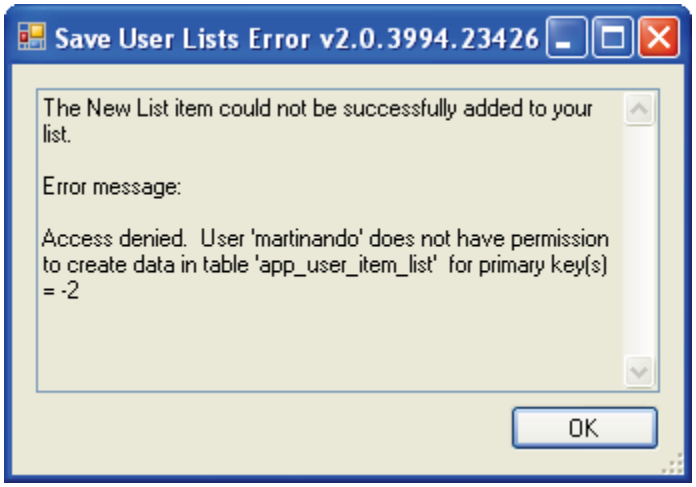
Action	Description
Select the **Enabled** checkbox	indicates that the user will be allowed to login to the Curator Tool
Select the **Active** checkbox	indicates the **UserID** is associated with an active cooperator – any data created or modified by this user will be tagged by his **CooperatorID**
Select a language for the user	The language setting determines what column headings, button text, etc. the user will see displayed in the Curator Tool
Assign the same **Site Code**	users who need to share lists within the Curator Tool must have the same **Site Code**
Add user to the **CT Users** group	If the user will be using the Curator Tool, he needs to be added to the **CT Users** group. (By default, a new user is added only to the **All Users** group – the user has no CT permissions at that point
Assign **All Access** permission to the user (if the user needs unlimited access)	Gives universal **Create/Read/Update/Delete** rights. The Administrators group has this permission as does the Administrator UserID. (Alternatively, you may decide to set up other Groups (see the next row in this table) with narrower permissions and not assign **All**
Add user to other groups as needed	Groups will have specific permissions which meet an organization's very unique needs. Groups are essentially templates for establishing permissions, so that each user does not need to be set up individually, but rather can be assigned to appropriate groups.

Error Messages

If security is enabled (the default situation), users not added to the **CT Users** Group the will receive several error messages when they log on. The following message displays when a user logs in to the Curator Tool and the user was not added to the CT Users Group:



The following error message will immediately be displayed as well, again for the same reason:

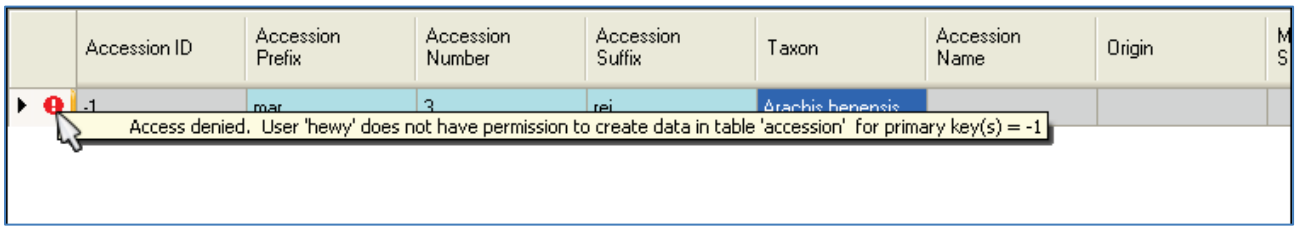


To correct this situation and to avoid the error messages, add the user to the **CT Users** group.

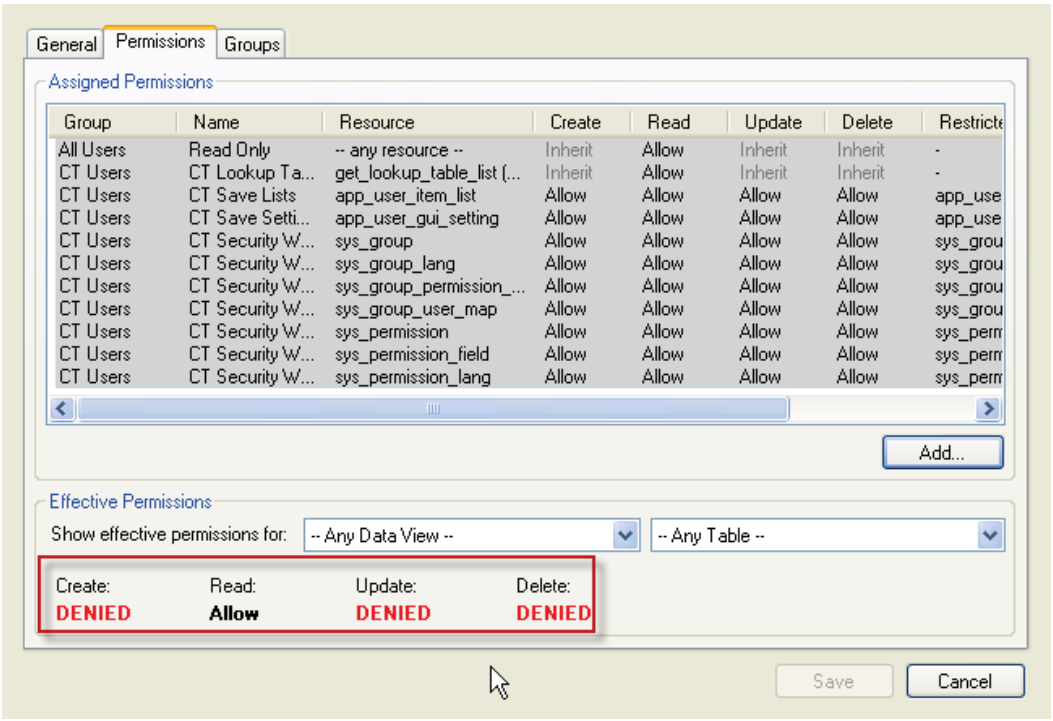
Establishing Groups and Permissions

However, even when users are included in **CT Users** , they still will not be able to save new records. This section explains why an organization will want to establish other Groups and Permissions beyond the **All Users** and **CT User** groups that come installed with GRIN-Global.

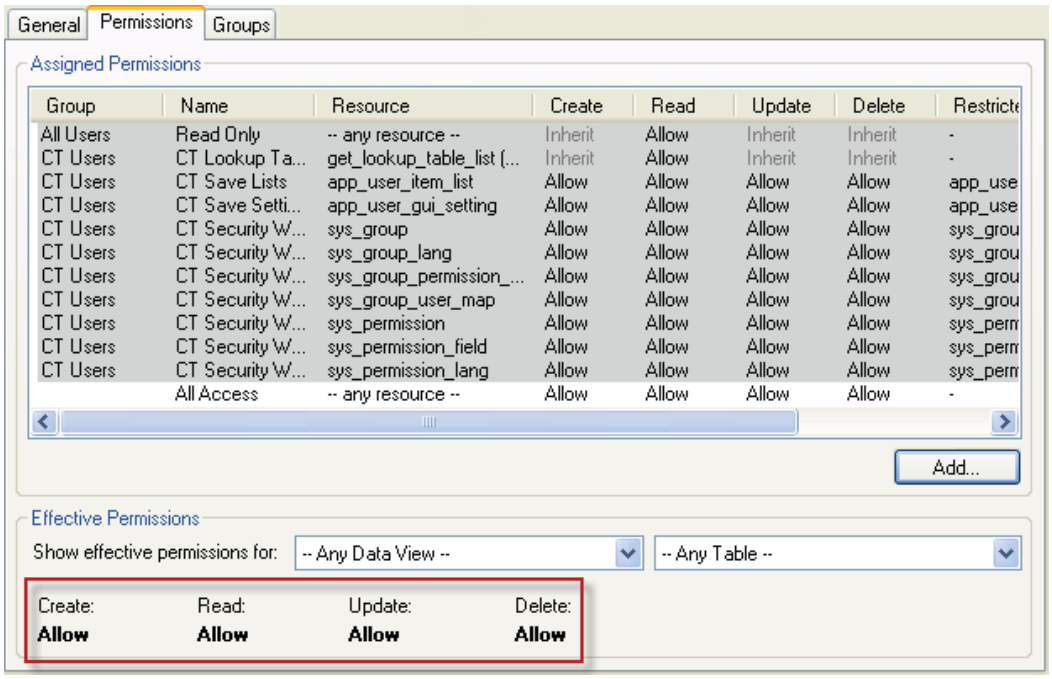
For example, a message similar to the following will be displayed after a failed Save action:



This user has not yet been given any permissions (other than “Read”) to the Accession (or any) table:

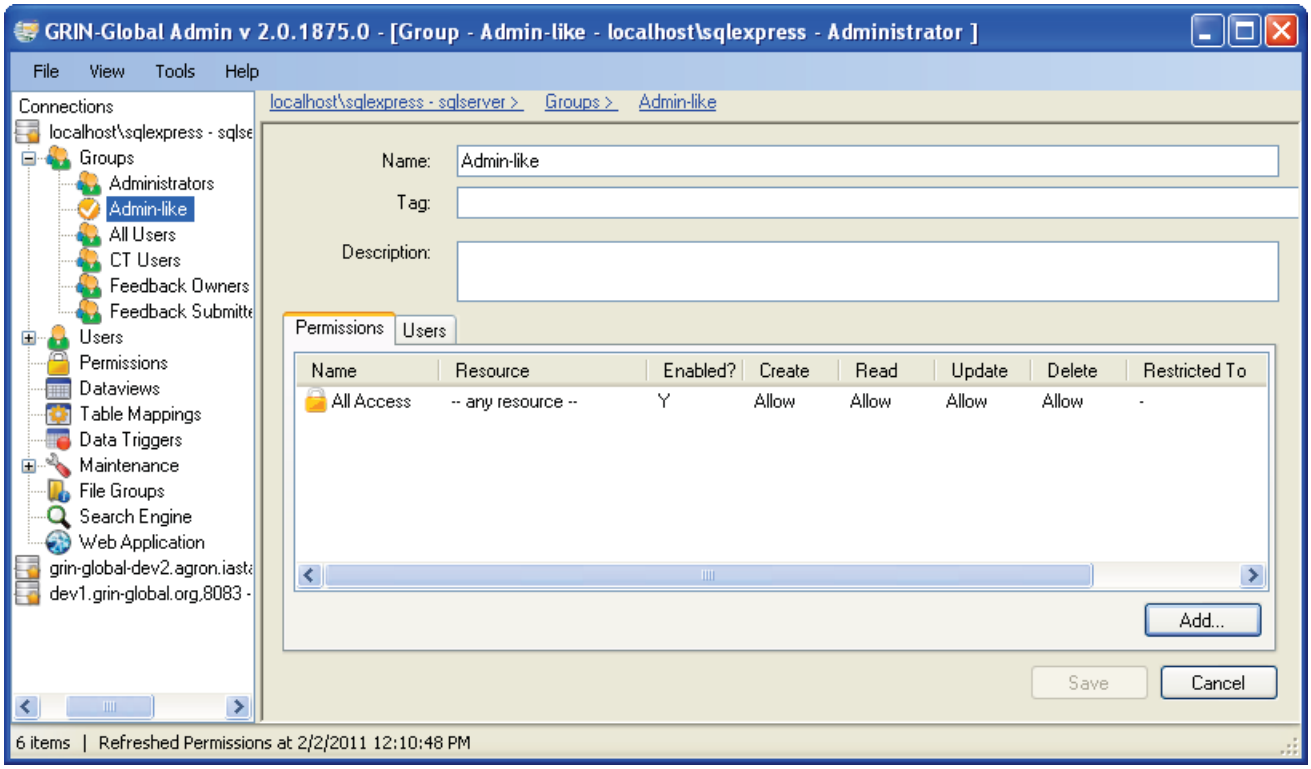


There are two quick methods for remedying this. One fix would be to add the user to the **Administrators** group (not typically done or recommended); a second method would be to grant the **All Access** (to any resource) permission to the user as shown in the following image:



However, a better solution would be to create a new group, modeled after the **Administrators** group, and then add users to that group. The advantages of doing this rather than simply adding users to the Administrators group are you keep the "true" Administrator permissions separate from other users.

In this example, the **Admin-like** Group is being created:



Any users then included in this group would initially have full data access as do administrators. Later, this group's permissions can be edited if necessary, for example to restrict access to certain data. The "true" administrators would not be impacted.

Also, groups or permissions can be established for certain categories of workers. For instance, an organization might want to establish that users who handle germplasm orders cannot modify accession records. Because of the flexibility with permissions, this can easily be accomplished.

Setting up a Site "Power User" for a Site

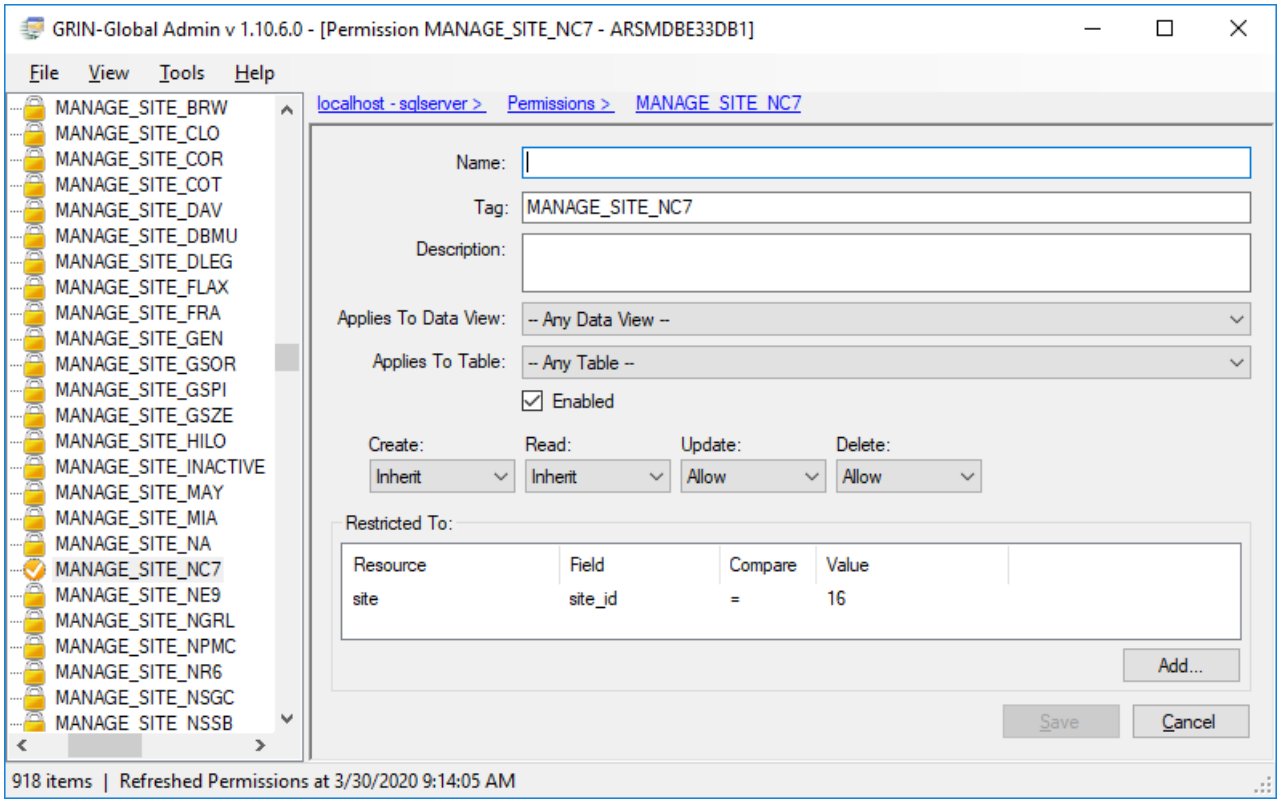
Two steps to follow to set up a user so that this user can essentially serve as a site administrator. He or she will be able to change ownership and permissions for his/her respective site.

1. Create a permission
2. Create a group using the permission and add user(s)

Create the Permission

Use the AT to create the permission. Give the permission a meaningful tag name, such as **MANAGE_SITE_NC7**. Set at least **Update** and **Delete to Allow**, possibly **Read** or even **Create** to if there is any doubt of that.

There are a few different ways to assign rights. If you want to give rights to any row in a particular table, change the "Applies To Table:" field from "-- Any Table --" to the appropriate table. If you want to have the rights apply to rows created by any user at a particular site, add a "Restricted To:" setting with **Resource = site**, **Field = *site_id***, Compare = "=" and Value = [the site id number]. Example:



Note that the two settings can be combined.

If you restrict to a site, but don't specify the table, the permission applies to that site's rows in any table.

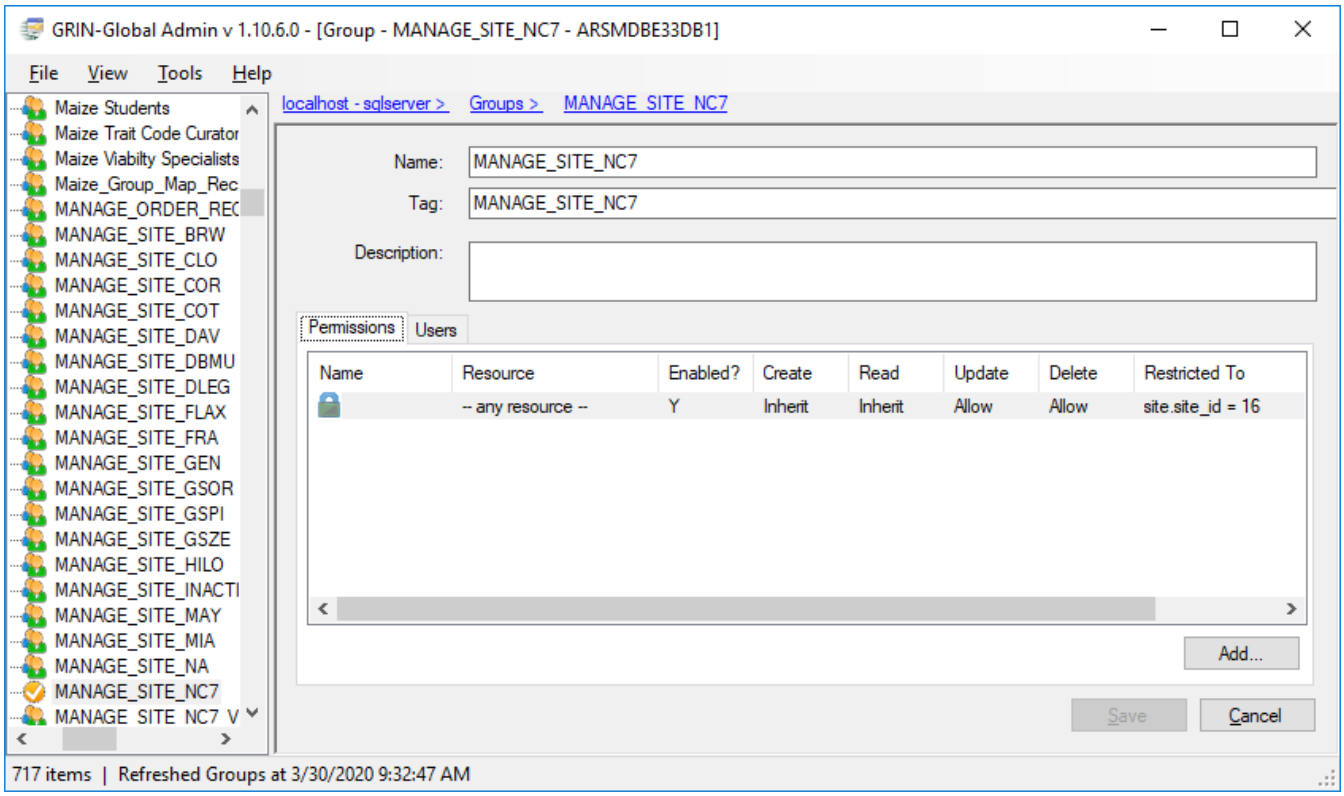
If you specify a table, but no site restriction, the permission applies to any row in that table.

If you specify both table and site, the permission applies to that site's rows in that table.

If you specify neither a table or site, the permission applies to all rows in all tables and becomes a full blown administrator permission (unless you are denying rights).

Create a Group Using the Permission and Add User(S)

Next create a group with an appropriate name (for example, use the same name as the permission). Add the permission in the **Permissions** tab and specify the user(s) who will receive the rights in the **Users** tab.



Appendix: Document Revision Notes

– November 22, 2022

- Added SQL example for displaying a user's permissions

– July 23, 2021

- corrected footer to include the page numbers
- also corrected some pagination breaks

– January 28, 2021

- corrected a typo
- added sample SQL and a link in the Ownership section to the Administrator Section