Ansys GRANTA MI 2021 R1

GRANTA MI
Access Control and Security Guide

Table of Contents

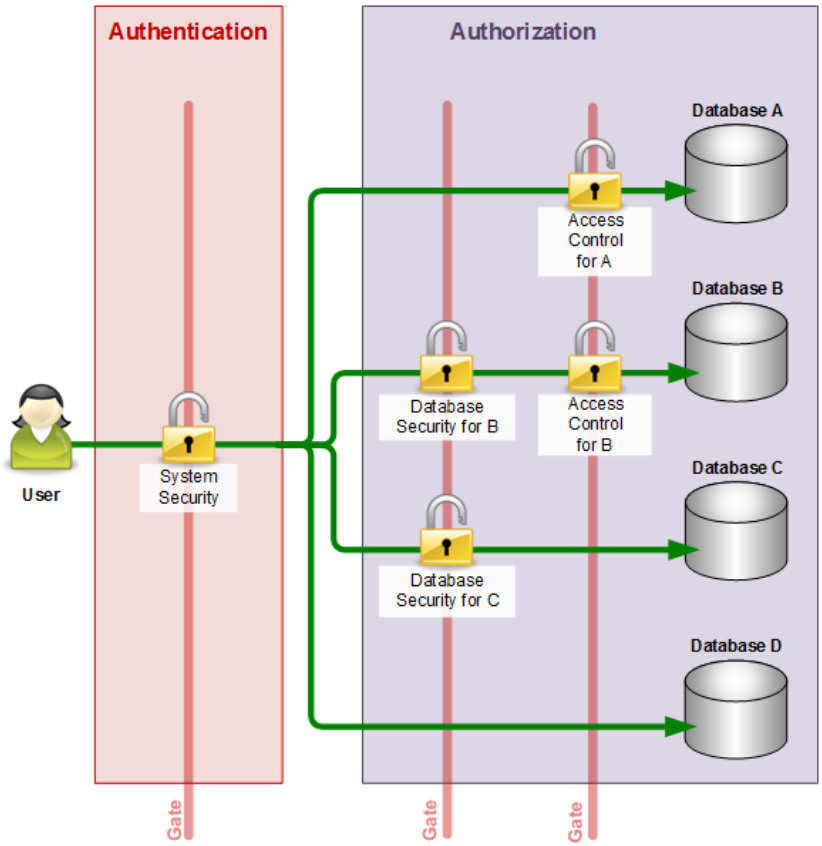
1 Introduction

User access to GRANTA MI can be considered as a system of 'gates':

1. *System Security*. Users are granted access the GRANTA MI system based on their membership of system security roles. Users can be authenticated with Windows Authentication or User Manager.
2. *Database Security (optional)* Users may be granted read, write, and administrative access to a particular database and the data in it, allowing users to be granted administrative access to a specific database without giving them administrative privileges to the whole GRANTA MI system.
3. *Access Control (optional)* This optional layer of security provides fine-grained control over who is allowed to access (view, edit) the individual tables, records, data, and attributes within a database.

These three levels of security act in combination. If no database security is set for a database, then system security is used to control access to that database. If database security has been configured, it takes precedence over system security settings for that database. Access control provides the most fine-grained security control.

Figure 1 Representation of GRANTA MI user access



Initial access to the system depends on the user's **system security role**.

Database security is set for Databases B and C, and so the user's privileges in those databases will be determined by her **database security role**.

Databases A and B also have access control applied. Access to the data in these databases is based on the user's **access control role** in that database.

Database D has no additional security, so the user's access to that database is determined entirely by her system security role

2 System security

Your choice of appropriate authentication and authorization options for your GRANTA MI system should be based on your particular deployment scenario.

Typically, authentication (determining the identity of a user before allowing the user to log on) is managed with Windows® Active Directory. Other identity providers can be used, including:

- **User Manager**, where users log in to GRANTA MI applications using account credentials that are defined in User Manager. See Section *2.3*.
- **OpenID Connect**, where GRANTA MI users authenticate against an identity provider system using industry standard OpenID Connect with OAuth 2.0 (OIDC). See Section *2.4*.
- **Custom authenticators**, where users are authenticated with software developed using the GRANTA MI SDK. See Section 2.5.

Authorization (determining the level of access that authenticated users have to resources in the GRANTA MI system) is role-based, with users and/or groups of users assigned to Granta system security roles. Different authorization options are supported:

- **Windows authorization**. AD groups are mapped to one of 5 Granta system security roles. Any user with Granta administrator privileges can map existing AD groups to roles, but AD group management (adding, removing users, creating new groups) is typically the responsibility of the IT department.
- **User Manager authorization**. Users are mapped to Granta roles using User Manager, Granta's role-based user access management application. Any user with Granta administrator privileges can create and manage Granta system user accounts, organize users into teams, and map teams and users to different roles.

2.1 Windows authentication and authorization

Where Windows is used for both user authentication and authorization in GRANTA MI:

- Users authenticate with their AD credentials
- A user's AD group memberships determine their role in GRANTA MI.

Users can log into GRANTA MI if they have an account in an Active Directory domain AND they are members of a Windows group that is mapped to one of the five system security roles, for example:

Windows security group	Granta security role
MI_READ	Read
MI_WRITE	Write
MI_POWERUSER	Power User
MI_GRANT	Grant
MI_ADMIN	Admin

During MI:Server installation, local Windows user groups are automatically created on the MI:Server host corresponding to these five roles, and the user performing the installation is automatically added to the MI_ADMIN Windows user group.

> **Note:** If MI:Server and MI:Viewer are installed on different computers, you will need to create Windows user groups on the **network** domain, as MI:Viewer needs direct access to the security groups on the MI:Server machine.

More groups can be mapped to Granta security roles after installation using the MI:Server Manager tool (**Security Groups>System Groups**), and additional database-specific or Access Control roles may also be defined within the GRANTA MI system. See the Help for MI:Server Manager for details.

Windows authentication configuration examples

In all the following examples, it is assumed that the Windows user groups have been created by a company IT system administrator. The task of the administrator of the GRANTA MI system is to map existing Windows user groups to Granta system security roles.

Example 1 Domain groups mapped to system security roles

In this example, Windows groups on the COMPANY network domain are mapped to Granta system roles. The groups are created on the network domain by the company system administrator.

System security role	Corresponding Windows group
Read	COMPANY\GRANTA_READ
Write	COMPANY\GRANTA_WRITE
Power User	COMPANY\GRANTA_POWERUSER
Grant	COMPANY\GRANTA_GRANT
Admin	COMPANY\GRANTA_ADMIN

Example 2 Local groups mapped to system security roles

In this example, MI:Server and MI:Viewer are installed on the same computer, allowing local Windows user groups to be mapped to the Granta system security roles. These Windows groups can be created by the (local) administrator of the computer on which MI:Server is installed.

System security role	Corresponding Windows group
Read	HOSTNAME\MI_READ
Write	HOSTNAME\MI_WRITE
Power User	HOSTNAME\MI_POWERUSER
Grant	HOSTNAME\MI_GRANT
Admin	HOSTNAME\MI_ADMIN

Example 3 Users from different network domains

Suppose your company has two geographic sites, London and Edinburgh, each with their own network domain. GRANTA MI has been installed at the Edinburgh site, but you want users on the London network domain to also have access.

To do this you would set up the GRANTA MI system security groups as normal on one network domain, the EDINBURGH domain.

In addition, your company system administrators need to set up a trust relationship between the LONDON and EDINBURGH domains. Then they should be able to add users from London into Edinburgh groups, or place London groups inside Edinburgh groups. This is all done as Windows administration outside of GRANTA MI.

2.2 User Manager authorization

User Manager may be used by any Granta administrator to manage role membership for authenticated users:

- Add users to your Granta system, and remove them.
- Organize users into **Teams** that reflect the business functions in your company, for example, *Metals*, *Polymers*, *Design*, *Simulation*. You can set up as many teams as you like, and you can easily add and remove users as needed.
- Assign Teams (and individual users) to the available **Roles** which determine what operations may be performed (view data, modify data, manage data, and so on). Roles may include system security roles, database security roles, and access control roles. Assigning a team to a Role results in all team members getting the greatest level of privilege as a result of combining their individual roles and team roles.
- Assign Teams (and individual users) to **Projects**. Projects are an optional feature that can be used to grant access to the data associated with a product or initiative as it progresses from concept through design and prototyping, production, testing, and release, depending on their access needs. Records associated with a project can only be accessed by members of that project; users who are not assigned to a project, or who are not members of a team assigned to a project, will not be able to see any records associated with that project.

2.3 User Manager authentication

In organizations with non-Microsoft systems, applications, or networks, where Active Directory is not an option, User Manager can also be used to authenticate GRANTA MI users. Granta user account credentials – login and password, and contact email address – are all defined in User Manager, and users log in to the system with their Granta account credentials.

2.4 OpenID Connection authentication

Support for OpenID Connect authentication is a Limited Availability feature introduced in GRANTA MI 2020 R2. This means that it is available for customers to use in production, but has limited support and documentation. Only a limited number of identity providers are supported in this release and there are additional configuration requirements to implement OIDC authentication as a Single Sign-On solution for GRANTA MI.

Contact Ansys Granta Technical Support for information on supported OIDC identity providers, and for configuration and setup documentation.

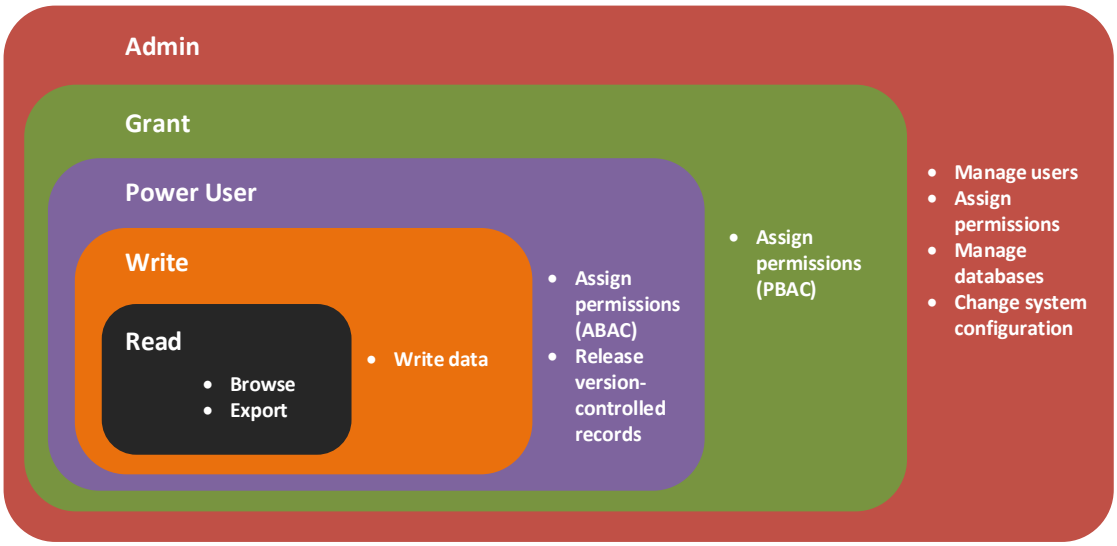
2.5 Custom authenticators

GRANTA MI can be configured to use custom (3rd party) authenticators. Supported configuration options include:

- Windows authentication and custom authorization (sometimes referred to as "Mixed mode"). See the *GRANTA MI Configuration Guide* for information on the necessary configuration for this.
- Custom authentication and authorization. Custom authenticators developed using the GRANTA MI SDK can be used to provide the rules for determining whether a user is authenticated and what they are authorized to do.

2.6 System security roles and privileges

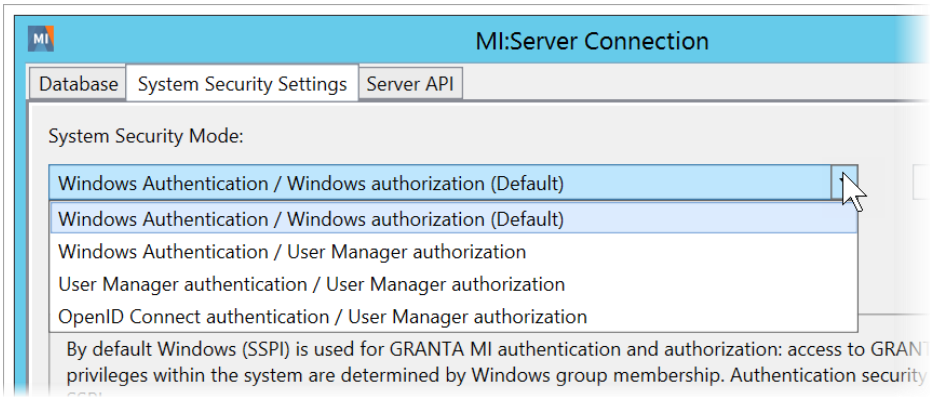
Membership of a system security role determines what users can do (their privileges) in the GRANTA MI system. The role system is hierarchical, with the Admin role having the maximum privileges, and the Read role having the fewest privileges. Higher roles include all the privileges of less privileged roles.



Where database security roles are defined, the privileges granted by the database role take precedence over privileges granted by a system security role.

2.7 System security configuration

To switch to a different system security model, use the MI:Server Connection tool:



Different system security configurations are supported:

- Window authentication and Windows authorization. This is the default configuration.
- Windows authentication and User Manager authorization.
- User Manager authentication and authorization.
- OpenID Connect authentication and User Manager authorization.

As well as these standard options, it is also possible to use custom authentication and/or authorization.

See the *GRANTA MI Configuration Guide* for information about configuring your system to use User Manager for authentication and/or authorization, and for information on using custom authenticators.

3 Database Security

Database security can be implemented for some or all of the databases in your GRANTA MI system, controlling who has read, write, and administrative access to a specific database. It can be used to allow users administrative access to selected databases without granting them administrative access to the whole GRANTA MI system.

Like system security, database security is role-based, with users assigned to a Read, Write, Power user, Grant, or Admin database security role. To gain access to the database, a user must therefore be a member of two roles—a system security role *and* a database security role.

Database security is optional. A set of roles can be configured for some or all databases in the system. If no database security roles are configured for a database, then the system security roles determine access to the database. When database security roles are set for a database, they take precedence over the system security roles for that database.

3.1 Privileges granted by database security role membership

Membership of a database security role determines what users can do (their privileges) in the database. The role system is hierarchical, with the Admin role having the maximum privileges, and the Read role having the fewest privileges. Higher roles include all the privileges of less privileged roles.

Table 1. Privileges granted by membership of a database security role

Database security role	Privileges
Read	Can browse and export data
Write	Can modify data
Power User	Can release version-controlled records Can modify attribute-based access control settings in the database
Grant	Has unrestricted access to the data in the database Can modify permission-based access control settings in the database
Admin	Can configure database security settings, manage (upgrade, lock) the database, edit the database schema, configure access control for the database

Users may have different privileges for system and database access; database role privileges override system role privileges. For example, a system Read user may have Admin privileges in a database where they have been assigned to a database Admin role. Conversely, a system Admin user may be assigned a less privileged (e.g. Read or Write) role on a database, and so in that database, they will be unable to perform any tasks that require a more privileged role.

3.2 Configuring database security roles

Database security is configured in the MI:Server Manager tool. See the Help for the MI:Server Manager tool for details.

3.3 Assigning users to database security roles

Windows Authorization

Up to five Windows groups should be mapped to the Read, Write, Power User, Grant, and Admin database security roles; all Windows users who wish to access the database must be a member of one of these mapped Windows user groups.

The Windows groups used for database security should exist on the network domain that contains the computer on which MI:Server was installed. If you do not have a network domain, you can use local groups on the computer on which MI:Server is installed, provided MI:Server and MI:Viewer are installed on the same computer. The domain of these local groups is the computer host name. Although you *can* map database security roles to correspond to non-existent Windows user groups, this is not recommended, as it adds time to the authentication process.

> **Note:** Granta recommends that you do not reuse Windows user groups for different purposes. For example, suppose you have created an MI_READ group to be mapped to the system security Read role, and you have added users to this group. You want the same users to have Write database privileges for Database Q. You should not reuse MI_READ for the database security Write role. Instead, you should create a new Windows user group, for example, DBQ_WRITE, and add the same users to the group.

User Manager Authorization

Teams and individual users can be assigned to database security roles in User Manager.

Assigning a team to a Role results in all team members getting the greatest level of privilege as a result of combining their individual roles and team roles.

4 Permission-based Access Control

A permission-based database access control system can be used to grant or deny access to tables, records, attributes and individual data items in the database based on read/write permissions set on those items; these permissions are mapped to roles in the access control schema.

4.1 The access control schema

The access control schema for a database defines *access categories* and *access permissions* that are applied in the database. For example, an access category called *Nationality* may have two permissions, *UK* and *US*; this means that all data in the database will have read and write flags for *UK* and *US*.

- Each permission in a schema has a set of read and write permissions.
- If a permission is defined in the schema, all data in the database will have a read and write flag for this permission. (Record links, Subsets, and Layouts do not have access control settings applied to them.)
- Categories do not have a hierarchy, i.e. one category is not ranked 'higher' than any other. Similarly, permissions do not have a rank.
- A schema for a database may have up to 128 permissions. This is the total number of permissions that have ever existed in the schema, for example, if 10 permissions are created and then deleted, the next permission created will be considered as the 11th permission.
- Only Admin and Grant users can change the permissions for a particular piece of data.

In the example schema below, the categories are *Nationality*, *Division*, and *Approval status*. The permissions for the *Nationality* category are *US* and *UK*:

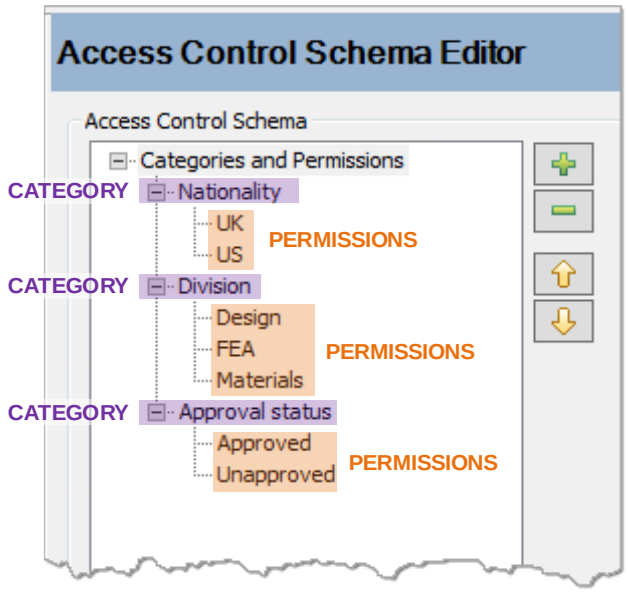
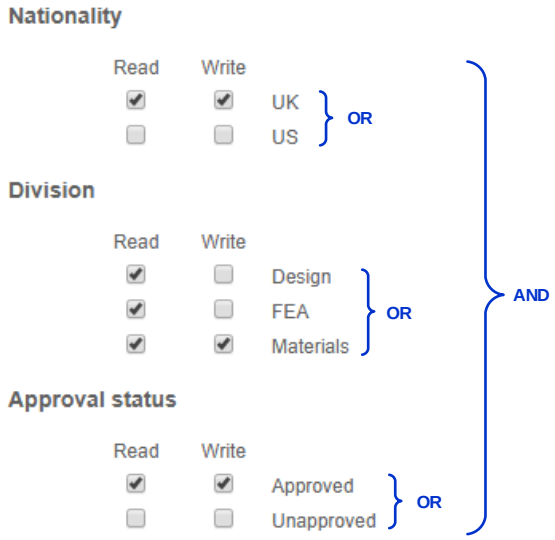


Figure 2. Example schema, as shown in the Access Control Schema Editor

If a schema consists of more than one category, the settings for an object are combined.

- Within a category, permissions are combined with OR.
- Between categories, permissions are combined with AND.



In the example above, in order to *view* the record, a user must have access control privileges in this database for:

UK **AND** [Designer OR FEA OR Materials] **AND** Approved

In order to *edit* the record, a user must have database access control privileges for:

UK **AND** Materials **AND** Approved

4.2 Access Control setting inheritance

Any new object added to the database inherits the access control settings of its parent.

The inherited access control settings are set at the time the new object is created. Once this has happened, if the settings on the parent change, these changes will not automatically be applied to the child objects (except for metadata).

- Volumes (Databases) do not inherit any access control settings.
- New tables inherit their access control settings from the volume.
- New records inherit their access control settings from their parent record. The root record in the tree is created with a set of default settings from the table.
- New attributes inherit their access control settings from the table.
- New data inherit their access control settings from both the record and the attribute to which the data corresponds. If the settings of one of the parents (record or attribute) are null, then the settings from the other parent are used. If both have values, the record and

attribute settings are combined with the binary AND operator to give the permissions for the data (see Figure 3).

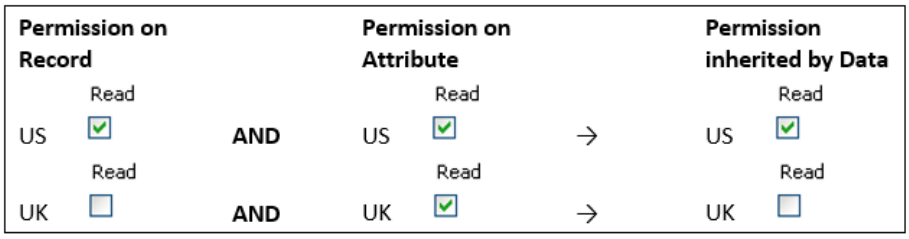


Figure 3. Record and attribute access control permission settings are combined with the AND operator to give the setting for data

Note that the operator in the example above is the binary (bitwise) AND. In a pair, the result is 1 if the first bit AND the second bit is 1. Otherwise the result is zero.

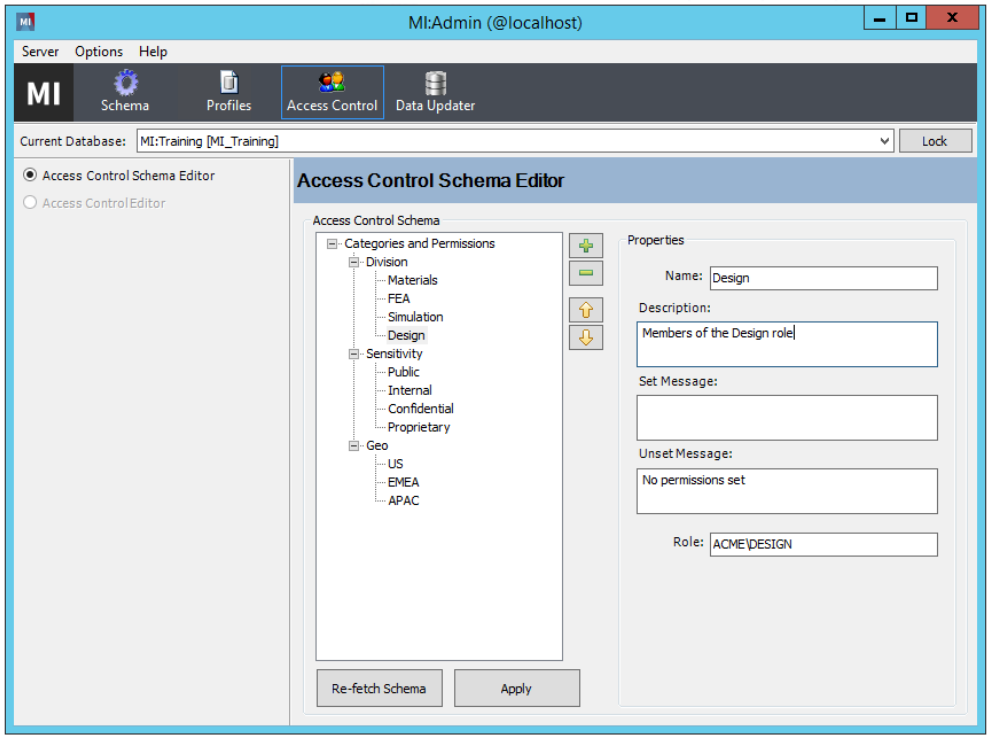
Metadata inherit their access control settings from both the meta-attribute and the data to which the metadata corresponds. It is not possible to edit the access control settings for metadata. If the settings for the parents are changed, then the settings for the metadata also change.

If you move or copy and paste a record, it will retain its access control settings from its old location (that is, the new location will not cause any changes to the access control settings).

4.3 Setting up permission-based access control

A permission-based access control schema is defined and modified using the Access Control Schema Editor in MI:Admin. You can add, modify, and delete categories and permissions, change the

mappings between access control permissions and roles, and define messages that are displayed in MI:Viewer when access-controlled items are viewed. See the MI:Admin Help for more information.



4.4 Permission-based access control with Windows authorization

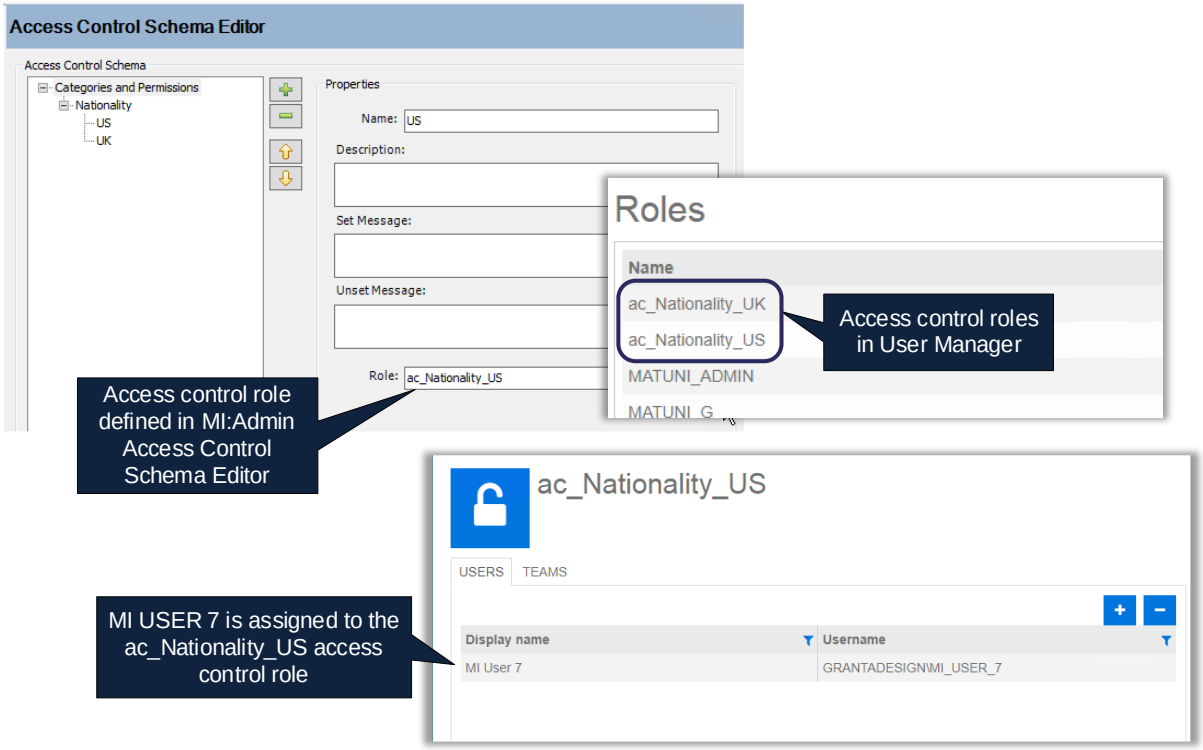
Each permission in the schema is associated with an access control role. Windows users must be a member of appropriate Windows security groups that are mapped to the relevant access control role.

In order to gain access to an access-controlled database, a user may therefore need to be a member of a number of different Windows groups:

- An AD group corresponding to a system security role
- (An AD group corresponding to a database security role, if database security has been configured)
- AD groups corresponding to access control schema roles

4.5 Permission-based access control with User Manager authorization

Each permission in the schema is associated with an access control role. Teams and individual users can be assigned to the access control roles, and removed from them, in User Manager:



4.6 Summary of permission-based access control

4.6.1 Viewing and editing data

In order for a user to view a piece of data

- the data must have the Read flag set for a particular access control permission.
- the user must have the requisite access control permission.
- the user must be in the Read system security role or higher.

In order for a user to edit a piece of data

- the data must have the write flag set for a particular access control permission.
- the user must have the requisite access control permission.
- the user must be in the Write system security role or higher.

4.6.2 User privileges

A Read user can

- view data for which they have the requisite access control settings (role membership and permissions).

A Write user can

- view and edit data for which they have the requisite access control settings.

A Power User can:

- view and edit data for which they have the requisite access control settings.
- release version-controlled records for which they have the requisite level of access control.

A Grant user can

- view or edit data in ALL access control permissions in the database.
- view or edit the access control permissions of ALL data.
- view or edit records with no access control settings.

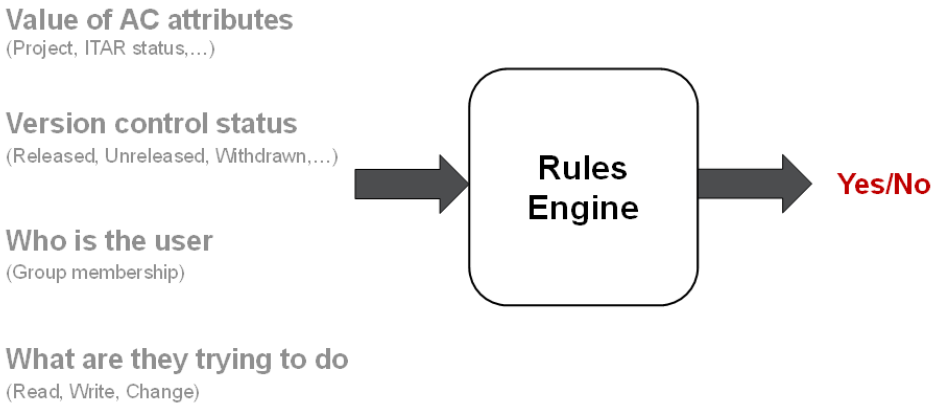
An Admin user can

- view or edit data in ALL access control permissions in the database.
- view or edit the permissions of ALL data.
- view or edit records with no access control settings
- view or edit the database access control schema.
- view or edit the database schema.

5 Attribute-based Access Control

With Attribute-based Access Control, access to database records is granted or denied based on the value of certain security attributes, called **Access Control Categories**, on those records.

Rules are used to determine, for a given record and user, what the various possible permutations of the Access Control Categories imply for that user in terms of their permissions to (a) Read the record, (b) Write the record and (c) Change the value of the Access Control Categories for the record. These 'Read—Write—Change' rules are embodied in a **Rule Engine**. A default rule engine is supplied with GRANTA MI.



GRANTA MI's Attribute-based Access Control system places data security in the hands of the MI Database Administrator rather than the IT System Administrator, reducing the need for numerous roles that are ad hoc and limited in membership. The combination of Access Control Categories and a powerful rule set that can evaluate different combinations of those attributes provides greater flexibility than a system based simply on numerous user roles and object permissions. Scenarios where this is desirable include:

- Where there are many possible combinations of properties for a record which determine visibility, but a relatively small number of roles is needed to cover them.
- When the visibility of data depends on the status of a record, and the rules for who can see and edit data will change as a record passes from one status to the next.
- Where users need an easy way to report on the access control for a collection of records, or where they need to search for records with particular permissions.

Attribute-based access control can be used with Windows authorization or User manager authorization.

5.1 Access Control Categories

Attribute-based Access Control relies on specialized security attributes ("Access Control Categories") which are defined in MI:Admin. For each possible value of each Access Control Category, a Rule Engine will determine which roles are able to read and write to the record, and change the Access Control Category value.

Each Access Control Category may comprise multiple discrete attributes (one per table), but all of them must be of the same discrete *type*, which may be single or multi-valued. For example, an Access Control Category discrete type called *Security Status* might have a single value, one of *Work In Progress*, *Production*, or *Archived*, while other Access Control Categories, for example, *Material Type*, could take multiple values.

Below are some examples of Access Control Categories.

- A ***Status*** attribute may be used to restrict who can see a material record based on a material's development status. For example, *Work In Progress*, *Production*, *Archived*.
- A ***Material Type*** attribute could be used to determine who can author data in the record, for example, only people in the *Composites* group can edit composite material records.

5.2 Rule engine

The GRANTA MI Rule Engine defines the access constraints for the whole GRANTA MI system, determining, for each Access Control Category, the roles that are able to read and write to the record, and change the value of its Access Control Categories.

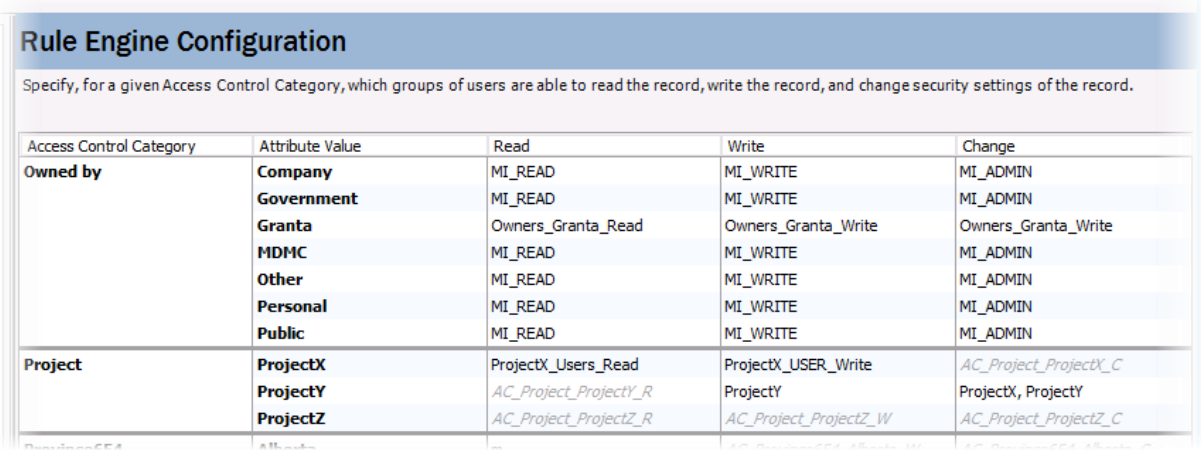
For example, when a user tries to access a record with the Access Control Categories defined in Section 5.1, the Rule Engine will have access to two pieces of information:

4. Is the user a member of any group associated with the *Material Type* for that record?
5. What is the value of the *Status* Access Control Category for this record?

Based on each combination of these values, the Rules Engine will determine whether the user can read and/or write to the record, and change the value of the Access Control Category.

5.2.1 Rule Engine configuration

Use the Rule Engine Configuration Tool in MI:Server Manager to define the access control rules for your system.



Rule Engine Configuration

Specify, for a given Access Control Category, which groups of users are able to read the record, write the record, and change security settings of the record.

Access Control Category	Attribute Value	Read	Write	Change
Owned by	**Company**	MI_READ	MI_WRITE	MI_ADMIN
	Government	MI_READ	MI_WRITE	MI_ADMIN
	Granta	Owners_Granta_Read	Owners_Granta_Write	Owners_Granta_Write
	MDMC	MI_READ	MI_WRITE	MI_ADMIN
	Other	MI_READ	MI_WRITE	MI_ADMIN
	Personal	MI_READ	MI_WRITE	MI_ADMIN
	Public	MI_READ	MI_WRITE	MI_ADMIN
Project	**ProjectX**	ProjectX_Users_Read	ProjectX_USER_Write	*AC_Project_ProjectX_C*
	ProjectY	*AC_Project_ProjectY_R*	ProjectY	ProjectX, ProjectY
	ProjectZ	*AC_Project_ProjectZ_R*	*AC_Project_ProjectZ_W*	*AC_Project_ProjectZ_C*

The Read, Write, and Change columns in the Rule Engine Configuration Tool show the roles permitted to access records with the specified Access Control Category value. Read, Write, or Power Users who are not in any of the roles specified in the **Read** column will not be able to see records with that Access Control Category value. (Admin and Grant users are always able to see the record.)

You can enter existing roles here, if suitable ones already exist. If you are using Windows authentication for your Granta system user authentication, the corresponding new AD groups will need to be created if they do not already exist – see Step 4 below.

Default role names, shown in grey, are autogenerated by concatenating the name of the discrete type used for the Access Control Category and its value as follows:

<*domain*>\AC_<*discrete_type_name*>_<*discrete_type_value*>_R|W|C

For example, a *Project* Access Control Category has the discrete type *Project*; this discrete type has three possible values: ProjectX, ProjectY, ProjectZ. The default roles for the Access Control Category are therefore:

- AC_Project_ProjectX_R
- AC_Project_ProjectX_W
- AC_Project_ProjectX_C
- AC_Project_ProjectY_R
- AC_Project_ProjectY_W

…and so on

WARNING Be aware that changing the name of a discrete type that is used in an access control category will affect the roles mapped to the access control category values in the Rule Engine. When the discrete type is renamed, **all** of the default role names in which it is used will automatically be renamed as well; in addition, any roles that have been entered manually in the Rule Engine configuration tool (overiding the suggested default name) **will also be overwritten** with the new default role name. This may affect users' access to data.

These default roles are useful where there is a large and potentially evolving collection of roles. For example, if you have an Access Control Category of *Project*, with hundreds of project values, increasing all the time, it is not necessary to configure your Rule Engine every time a new project is added. Overriding the suggested roles with specific roles, is useful when the many permutations of Access Control Category values map on to a relatively small number of system roles, which change rarely.

Note that any changes to the Rule Engine will require a GRANTA MI service restart afterwards.

5.3 Implementing attribute-based access control: a summary

The steps required to set up attribute-based access control are as follows.

Step 1: Enable Attribute-based Access Control in your GRANTA MI system

1. Open MI:Server Manager.
2. Click **Access Control Settings** in the navigation pane on the left of the window.
3. Click **Attribute-based** and choose the rule engine to use, for example, DefaultAccessControlRuleEngine. (Unless you have developed your own rule engine, this will be the only option available.)
4. Click **OK**. MI:Server must be restarted.

Step 2: Define the Access Control Categories and discrete types that are needed

Create the discrete types and Access Control Categories you will need for each database:

1. Open MI:Admin and click on **Schema**.
2. Click **Edit Discrete Types** and add the required discrete types. For example, a discrete type called *Status* might have the values “Work In Progress”, “Production”, and “Archived”; a discrete type called “Export Control” that takes a single value “Controlled”.
3. Click **Edit Access Control Categories** and add new Access Control Categories. For example, you could create a *Current Status* category with the discrete type *Status*; an *Export Control Status* category with the discrete type *Export Control*.
4. Map your Access Control Categories onto one or more attributes within the tables in your database. If the attributes do not exist, you can create them directly in the interface. You may only select pre-existing attributes with the discrete type chosen in step 3.
5. Add the Access Control Attributes to layouts.

   Like other attributes, Access Control Attributes will not be visible/editable in MI:Viewer or MI:Explore unless they are added to layout(s). Layouts are defined per-table in MI:Admin; see the MI:Admin help for details. You may also want to add a layout heading, such as “Security Settings”, under which Access Control Categories are displayed.
6. Repeat steps 2-5 for additional databases.

Step 3: Configure the Access Control Rules in the Rule Engine

Specify, for a given Access Control Category, which roles are granted read and write access, and are able to change the value of the Access Control Category.

1. Open MI:Server Manager.
2. Under **Access Control Settings** in the navigation pane on the left of the window, click **Rule Engine Configuration**.
3. Enter roles in the **Read**, **Write**, and **Change** columns for each possible value of the Access Control Category, or use the suggested default role names.

   You can select multiple cells and type once to fill them all, and you can quickly copy/paste between multiple cells. Use commas to separate multiple values in cells.
4. Click **Save** when you are done.

Step 4: assign users to any new roles

If additional roles have been created for particular combinations of Access Control Category and access level, you will need to assign users to those roles:

- Windows authentication: this means creating new Windows AD groups and adding the relevant domain users to them.
- User Manager authentication: the new roles will appear in User Manager, where you can assign Teams and individual users to them.

Step 5: Set Access Control Category values for database records

When the access control system setup has been completed (Steps 1 through 4 above), you can now start using it by assigning Access Control Category values to records in your database.