

Administering users and groups

{scrollbar}

INLINE

You can add users and groups via the Geronimo Administration Console or by modifying some configuration files.

We will start simple by using the realm provided by Geronimo by default. Then, as we explore the different realms and security configurations, we will come back and revisit some of the topics as needed.

To manage users and groups via the Geronimo Administration Console the **Users and Groups** portlet is available on the **Console Navigation** menu on the left hand side. Here you will find two portlets, one for administering users and another for administering user groups, both are illustrated in the following figures.

Console Realm Users [Window Controls]

[Create New User](#)

Username	Actions
system	Edit Delete

Console Realm Groups [Window Controls]

[Create New Group](#)

Group Name	Actions
admin	Edit Delete

To change a user's password click on **(Details** next to the user you want to update in the **Console Realm Users** portlet, it will bring up the UserID and Password so you can update that profile.

Console Realm Users [Window Controls]

UPDATE USER

UserID: system

Password:

Confirm Password:

To remove a user click on the corresponding **Delete**, you will be prompted to confirm deletion of that user, click **OK**.

To add a new user click on **Create New User**, you will be prompted for a UserID and Password (twice), enter those values and click **Add**.

Console Realm Users [Window Controls]

ADD USER

UserID: user1

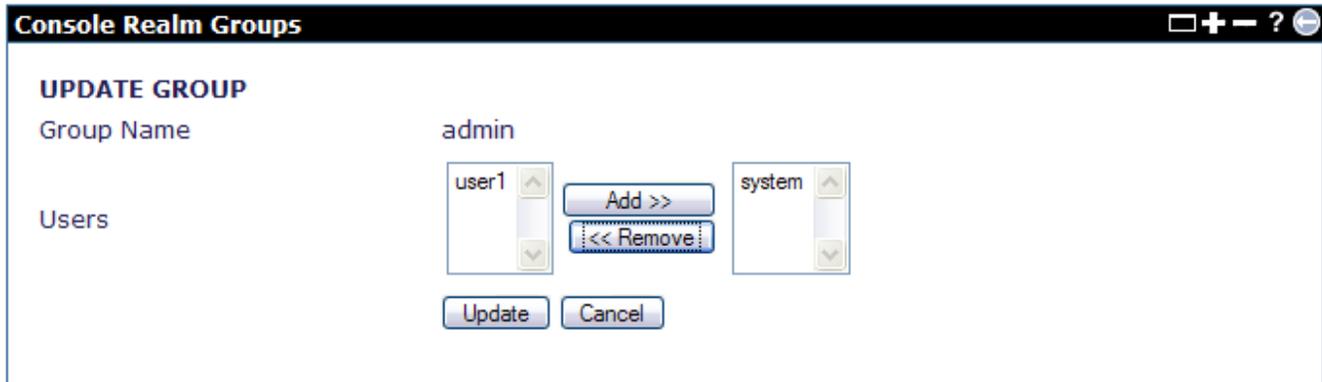
Group: admin [v]

Password:

Confirm Password:

Once you created new users you can add them to group. By default, the group **admin** is available and the user **system** is in that group. If you click on the **Details** next to the **admin** group you will see the user **system** in the window on the right and any other available user will be listed in the window on the left.

To add a new user to this group select the user first, then click **Add >>** and then click **Update**.



To create a new group click on **Create New Group**, this step is very similar to the one mentioned before for the users. In addition to be prompted for adding users to this group you will also have to provide a group name. Once you entered the new group name and added the users click on **Add** to finish.

The changes you made via the **Console Realm Users** and **Console Realm Groups** portlets are reflected in two different files, these files are **users.properties** and **groups.properties** respectively and they are located in the `<geronimo_home>\var\security` directory.

You can equally administer users and groups by modifying directly these files:

- `users.properties`
- `groups.properties`

`users.properties` uses the `<user_name>=<password>` format, `groups.properties` uses the `<group_name>=<user_name>` format. See the following examples for additional details.

```
solidusers.properties system=manager user2=password user1=password
```

As we are using the basic, by default, security configuration you will see the user IDs and passwords are stored in plain text. You can add, remove and change passwords from this file.

```
solidgroups.properties admin=system,user1 users=user2
```

Just like with the users, with the `groups.properties` you can add and remove groups and users to those groups.

The files mentioned in this sections along with the all the security configuration in addition to user names and passwords are defined in the **geronimo-properties-realm** security realm covered in the [Administering security realms](#) section.

Changing password for derby Adminchangingpasswordforderbyadmin

Starting from G217, a new user **dbadmin** is defined to manage all connections to the embedded Derby server. By default, the user and its group are defined in the `/var/security/groups.properties` file, which you should NOT update. But you can update the default password *manager* to any other combinations you like. While doing so, you must follow the steps below to make sure the whole server and relevant database pools work well after you updated the password.

1. Start Geronimo server
2. Update the password of **dbadmin** via **console >security > Users and Groups** portlet
3. Edit the user password in existing datasources via **console > Services > Database pools** portlet especially for the system related datasources
4. Shutdown Geronimo server
5. Update the **userPassword** attribute of DerbyNetwork GBean in `var\config\config.xml` with the new password
6. Restart Geronimo server

Controlling Derby authenticationcontrollingderbyauthentication

Starting from 2.2.1, a system property **derby.connection.requireAuthentication** is supported to control the switch of Derby authentication on a Geronimo server. You can set the property to *true* to enable derby authentication by passing the property and its value to **GERONIMO_OPTS** as followed before the server is started, and set to *false* to disable. By default, the value is *true*.

- non-Windows systems **export GERONIMO_OPTS=-Dderby.connection.requireAuthentication=true|false**
- Windows systems **set GERONIMO_OPTS=-Dderby.connection.requireAuthentication=true|false**

Accessing user-defined Derby databasesaccessinguserdefinedderbydatabases

When you're using the Apache Derby database, a table is always in a schema. If you don't specify a schema explicitly, Derby implicitly uses the built-in *app* schema. A second built-in schema called *sys* which is used to isolate system tables. If you specify a user name *A* when creating the database, thinking about the scenario you are [deploying a datasource](#) and set the **create database** parameter as **true**, then you create a table *T*; the fully qualified name of the table *T* will be *A.T*.

When executing SQL commands upon a Derby database, always remember to specify the fully qualified name of the table. Otherwise, you might be experiencing certain `SQLExceptions` such as `Connection authentication failure`, especially when the username and the schema are not the same.

If you turn the derby authentication on and you still want to access the databases you created, make sure that the user name is defined in **derbyadmin** group within the `groups.properties` file, and the user's password defined in `users.properties` files as followed.

```
solidgroups.properties
admin=system,
derbyadmin=dbadmin,app,user1
solidusers.properties
system=manager
dbadmin=manager
app=app
user1=password
```

where

- *dbadmin* is the default derby system user.
- *app* is the user name to access a table with the default *APP* schema.
- *user1* is name of the user who creates the database.
- *password* is the plain text password of user **user1**.

Note that if a database is created via **Create DB** button on the **DB Manager** portlet, the default schema is *DBADMIN* and its default owner is **dbadmin**.