

Administering security realms

[← Administering certificates](#)

[↑ Configuring security](#)

[Administering users and groups →](#)

To administer security realms via the Geronimo Administration Console the **Security Realms** portlet is available on the **Console Navigation** menu on the left hand side. This portlet allows you to add a new security realm or edit an existing one. To remove realms you would normally use the command line option with the Deployer tool.

Security Realms [view]

This page lists all the available security realms. Server-wide security realms can be edited, while security realms deployed as part of a single application cannot (change the deployment plan in the application instead).

For each realm listed, you can click the **usage** link to see examples of how to use the realm from your application.

Name	Deployed As	State	Actions
geronimo-admin	Server-wide	running	edit usage

[Add new security realm](#)

Listed in this portlet you will find all the available security realms. By default, the security realm used by Geronimo to authenticate users via properties file is **geronimo-admin**.

When you edit an existing realm (in this case geronimo-admin) you will be presented with the following screen, note that you will not be able to change the realm name nor the login domain name.

Security Realms

[\[view\]](#)

This page edits a new or existing security realm.

A security realm may have one or more login modules. Many simple realms have only one login module. Additional login modules may be used to access more underlying security information stores, or to add functionality such as auditing to a realm without affecting the authentication process for the realm.

Realm Name: geronimo-admin

A name that is different than the name for any other security realms in the server (no spaces in the name please). Other components will use this name to refer to the security realm.

Login Module 1**Login Domain Name: geronimo-admin**

The login domain for this login module, which must be unique among all modules in the security realm. This can be used to distinguish principals from two otherwise identical login modules (for example, from two LDAP login modules pointing to two different LDAP servers)

Login Module Class: org.apache.geronimo.security.realm.providers.PropertiesFileLog

The fully-qualified class name for the login module.

Control Flag: Required

The control flag for the login module, which controls what happens to the overall login processing if this login module succeeds or fails. For more information see [javax.security.auth.login.Configuration](#).

Support Advanced Mapping: No

Normally Geronimo can't distinguish between two different principals that have the same name and same principal class but were produced by two different login modules. If this option is enabled, Geronimo will "wrap" principals to track which login module and realm each principal came from. This lets you use the "realm-principal" and "login-domain-principal" elements in your security mapping in Geronimo deployment plans.

Configuration Options:

```
usersURI=var/security/users.properties
groupsURI=var/security/groups.properties
```

Any configuration options necessary for the login module, in the standard Java properties format (one per line, name=value)

[Cancel](#)

The following example illustrates the deployment plan generated by this realm.

geronimo-properties-realm

```
<module xmlns="http://geronimo.apache.org/xml/ns/deployment-1.2">
  <environment>
    <moduleId>
      <groupId>console.realm</groupId>
      <artifactId>geronimo-admin</artifactId>
      <version>1.0</version>
      <type>car</type>
    </moduleId>
    <dependencies>
      <dependency>
        <groupId>org.apache.geronimo.configs</groupId>
        <artifactId>j2ee-security</artifactId>
        <type>car</type>
      </dependency>
    </dependencies>
  </environment>
  <gbean name="geronimo-admin" class="org.apache.geronimo.security.realm.GenericSecurityRealm" xsi:type="dep:
gbeanType"
    xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.2" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance">
    <attribute name="realmName">geronimo-admin</attribute>
    <reference name="ServerInfo">
      <name>ServerInfo</name>
    </reference>
    <xml-reference name="LoginModuleConfiguration">
      <log:login-config xmlns:log="http://geronimo.apache.org/xml/ns/loginconfig-2.0">
        <log:login-module control-flag="REQUIRED" wrap-principals="false">
          <log:login-domain-name>geronimo-admin</log:login-domain-name>
          <log:login-module-class>
            org.apache.geronimo.security.realm.providers.PropertiesFileLoginModule
          </log:login-module-class>
          <log:option name="usersURI">var/security/users.properties</log:option>
          <log:option name="groupsURI">var/security/groups.properties</log:option>
        </log:login-module>
      </log:login-config>
    </xml-reference>
  </gbean>
</module>
```

As we mentioned before, this plan is for the default, properties file based, security realm. When you create a new realm you will have to choose from the following realm types available:

- Certificate Properties File Realm
- Database (SQL) Realm
- LDAP Realm
- Properties File Realm
- Other

The last available option lets you create your custom realm type when none of the above fits your environment needs.

Having the **Properties File Realm** covered by default we will now focus on the other alternatives.

- [Certificate Properties File Realm](#)
- [Database \(SQL\) Realm](#)
- [LDAP Realm](#)