

---

## Article

[Tani Frankel](#) · Feb 25, 2016 2m read

# Manipulating Security Entities Programmatically

## Question:

How can I create, change etc. Security entities (like Users and Roles) programmatically?

## Answer:

You can use the [Security](#) package classes in the %SYS namespace. For example [Security.Roles](#) or [Security.Users](#).

Of course in order to perform these actions the user will require the needed authorization.

Here's a small example using Security.Roles:

```
// Create a Role
%SYS>Set status = ##class(Security.Roles).Create("testRole","a test Role","%Development:U,%DB_USER:R")

// Creation was successful
%SYS>Write status
1

// Now lets examine the Role created
%SYS>Set status = ##class(Security.Roles).Get("testRole",.properties)

// We got the properties correctly
%SYS>Write status
1

// The properties were returned by reference
%SYS>ZWrite properties
properties("Description")="a test Role"
properties("GrantedRoles")=""
properties("Resources")="%DB_USER:R,%Development:U"

// Now we want to add another permission
%SYS>Set properties("Resources")=properties("Resources")_"",%DB_SAMPLES:RW"

// And modify our Role
%SYS>Set status = ##class(Security.Roles).Modify("testRole",.properties)

%SYS>Write status
1

// Let's verify this worked
```

```
%SYS>Set status = ##class(Security.Roles).Get("testRole",.latestProperties)
```

```
%SYS>ZWrite latestProperties
```

```
latestProperties("Description")="a test Role"
```

```
latestProperties("GrantedRoles")=" "
```

```
latestProperties("Resources")="%DB_SAMPLES:RW,%DB_USER:R,%Development:U"
```

Note: Be aware that the current behavior (as of v2016.1) is that if the comma-delimited Resource list includes an empty element (e.g. 'aaa,,bbb') the addition of resources will stop there.

This behavior has been logged to be examined if desired, and if not will be fixed.

[#Security](#) [#Caché](#)

---

Source URL: <https://community.intersystems.com/post/manipulating-security-entities-programmatically>