NIST RBAC PHP API

Chapter 1: Introduction

This document describes the background of the NIST RBAC PHP API. RBAC stands for Role-Based Access Control and it is one of the most prevalent forms of authorization in applications. Rather than assigning permissions directly to users they are bundled into roles and roles are assigned to users. This has the advantage that when anything changes in the permissions (permissions added, deleted or changed) this doesn't impact the users directly as they are assigned to roles and that assignment doesn't change.

The NIST RBAC PHP API offers a method of applying Role-Based Access Control to your PHP application. For your application you only need to use one API call: the CheckAccess function. All the other functions in the API are used for managing the various users, roles and permissions but that can be done separately from your application. This document will explain how to use the API for your application.

The API is based on the NIST RBAC standard for role based access control. It implements the "Basic RBAC Model" as defined in "A Proposed Standard for Role-Based Access Control[1]. This is for the largest part compliant with the "Flat RBAC" definition as described in the approved standard "The NIST Model for Role-Based Access Control: Towards a Unified Standard".

Why choose a draft standard?

The reason for choosing to implement the API based on the draft instead of the formal standard lies in the fact that the draft document specifies the actual API calls in a formal notation (Z) thereby offering a level of depth and clarity for implementation that is missing from the formal standard.

What's different?

Analysis of both documents, specifically the "Basic RBAC Model" versus "Flat RBAC Model" shows one major difference; the usage of the Session object. In "Flat RBAC Model" this is missing, only to resurface in the "Constrained RBAC - Dynamic Separation of Duties" model. In the draft "Basic RBAC Model" the Session is included right from the start. The Session object allows you to dynamically (during the user session) assign and remove roles. The session roles are not permanent, after the user logs out or the user has been logged out due to session time constraints the dynamically (temporarily) assigned roles are removed.

Besides this rather attractive feature of being able to assign or remove roles during a session it is also common for many developers to think of sessions when developing secure web applications. To those developers the Session object will speak out as it allows them to tie the Session object in with the session methodology they use for their web application.

There are two functions directly associated with the session role functionality: AddActiveRole and DropActiveRole. If you choose to ignore these two functions than by all means your application complies with the "Flat RBAC Model" of the formal standard.

[1] ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001.

With this in mind I have opted to implement the draft "Basic RBAC Model" rather than the more simple (and constrained) "Flat RBAC Model".

What's in the package?

This software distribution contains three major items:

- A physical data model in the form of DDL instructions for MySQL. It has been tested with MySQL Server 5+, older versions are not supported but might work
- A web based RBAC Management application that allows you to manage users, roles and permissions. It's been designed to work with PHP 5 and it uses PHP 5 features so it will not work on PHP 4 based installations.
- The NIST RBAC PHP library that offers API calls for your application. This is also designed to work with PHP 5 as it makes use of the mysqli extension.

Added to the distribution is a simple demo application to showcase usage of the API. Bear in mind that the NIST RBAC PHP API is also applied to the RBAC Management application itself, so that is a demo as well (albeit a rather complicated one, hence the addition of a simple demo).

A test framework is available that tests all the API calls and reports back to the user whether the calls are executed successfully.

Full PHPDoc API documentation is included with the distribution.

It's not OO!

No it's not OO, I don't like OO that much and especially not in PHP. If you want to make it OO it should be relatively straightforward as each function is completely isolated and there are no globals (some constants are used for setting up the database connection and file paths but they should not cause any problems). So if you absolutely must make this OO for your own peace of mind go right ahead, it shouldn't be too arduous. If there's anything in the code I need to change to facilitate usage in an OO scenario then drop me a line and I'll see what I can do to incorporate those changes in future versions.

Chapter 2: Getting started

Getting the NIST RBAC PHP API working for your application should be easy. There are a couple of steps involved that are documented in this chapter.

Prerequisites

The software has a small number of prerequisites in order to run successfully:

- MySQL Server 5+
- PHP 5+
  - mysqli extension (database access, prepared statements, bound results)
  - mbstring extension (for Unicode handling)
  - filter extension (input checking)
  - ctype extension (input checking)
  - sessions extension (session management)
  - hash extension (password hashing)
  - json extension (internationalization)

Step 1: The database

The software distribution contains a SQL import file that needs to be executed. You can find the file import.sql in the install folder of the distribution. Open your favorite database management tool or a direct connection to the mysql server and create a database with utf8_unicode collation. You can name it whatever you like just a long as you configure the database correctly in the library at a later stage. I would advise you to create a separate user and password for accessing the RBAC database. It can work with limited rights, as long as you allow SELECT, INSERT, UPDATE and DELETE you should be fine.

After you have created the database (for the sake of this tutorial let's call it rbac), you can import the import.sql file. If everything went fine you should have the following tables in your rbac database:

- `object`
- `operation`
- `permission`
- `role`
- `role_permission`
- `session`
- `session_role`
- `user`
- `user_role`
- `user_session`

You can check whether the import was successful by executing the following query:

```
SELECT role.name AS Role, permission.name AS Permission, object.name AS
Object, operation.name AS Operation
FROM role_permission
INNER JOIN permission USING (permission_id)
INNER JOIN object USING (object_id)
INNER JOIN operation USING (operation_id)
INNER JOIN role USING (role_id)
WHERE role_id = (SELECT role_id FROM role WHERE name = 'Administrator')
```

This should give you a list of all permissions, objects and operations associated with the 'Administrator' role. There should be 50 rows if the query is run successfully.

Step 2: The NIST RBAC PHP API

Now that your database is installed and you (hopefully) have created a dedicated user for it, it is time to configure the NIST RBAC PHP API.

Open the file configuration.php in the /lib folder in your editor. At the top of the file there is a configuration section (indicated by START CONFIGURATION SECTION and END CONFIGURATION SECTION) that allows you to configure the following items:

- DATABASE_SERVER: [*ip-address | alphanumeric*] use the IP address or URL of the database server
- DATABASE_USER: [*alphanumeric*] fill in the database user used for the rbac database
- DATABASE_PASSWORD: [*alphanumeric*] fill in the password used for the rbac database
- DATABASE_NAME: [*alphanumeric*] fill in the name of the database, in our example it is called rbac
- DATABASE_PORT: [*numeric*] fill in the port number of the database server, by default this is 3306
- INACTIVE_SESSION_TIMEOUT: [*seconds*] fill in the number of seconds of inactivity before the session is deleted
- TOTAL_SESSION_TIMEOUT: [*seconds*] fill in the number of seconds a session is allowed to live before it is deleted

Please change the values to match your own environment. Make sure that you have verified the settings and have tested them with a real database connection before you commit them to this configuration section in order to avoid troubleshooting later down the road.

Step 3: Run the test framework

To verify whether you have configured the database and the NIST RBAC PHP API correctly you have to run the test framework. Before this can be done it needs to be configured with the correct URL and file paths. There is a configuration.php file in the test folder of the distribution. Open the configuration.php file with a text editor. The file contains the following entries:

- PATH: [*url*] The abs path of the test framework where the leftmost slash (/) is the webroot

- INCLUDE_PATH: [*url*] The abs path of the include files (template, css, js)
- LANGUAGE: [*language code*] The name of the language file to load
- STATUS: [DEVELOPMENT/PRODUCTION] Status of the environment, determines error logging and reporting
- ERROR_LOG_PATH: [*file path*] Where the error log is written
- TIMEZONE: [*UNIX timezone*] Fill in the timezone information (for logging purposes)
- TITLE: [*alphanumeric*] This is the title of the page
- SUBTITLE: [*alphanumeric*] This is the subtitle of the page

Change the values according to your own configuration. If you have placed all the files in the webroot of your webserver and you have kept the folder names identical to the distribution you should have the following configuration.php contents:

```
defined('PATH') or define('PATH', '/apps/test');
```

The other items of the configuration.php file have been left out because they do not directly impact runtime behaviour.

Start up your browser and invoke the url of the test framework. If you have copied over the files in the webroot of your webserver this should be: `http://www.yourserver.org/apps/test/`

The tests will run automatically and inform you of success or failure.

If the tests have run successful it's time to go to the next step.

Step 4: Configure the NIST RBAC Management Application

The NIST RBAC Management Application can be used to manage users, roles, permissions and objects for your application. It offers a web GUI on the NIST RBAC data set offering near 100% coverage of all the NIST RBAC PHP API calls.

The manager application can be found in folder /apps/manager. There is a configuration.php file in this folder. Open the configuration.php file with a text editor. The file contains the following entries:

- PATH: [*url*] The abs path of the manager application where the leftmost slash (/) is the webroot
- INCLUDE_PATH: [*url*] The abs path of the include files (template, css, js)
- LANGUAGE: [*language code*] The name of the language file to load
- STATUS: [DEVELOPMENT/PRODUCTION] Status of the environment, determines error logging and reporting
- ERROR_LOG_PATH: [*file path*] Where the error log is written
- TIMEZONE: [*UNIX timezone*] Fill in the timezone information (for logging purposes)
- TITLE: [*alphanumeric*] This is the title of the page
- SUBTITLE: [*alphanumeric*] This is the subtitle of the page

Change the values according to your own configuration. If you have placed all the files in the webroot of your webserver and you have kept the folder names identical to the distribution and you're hosting on Unix you should have the following configuration.php contents:

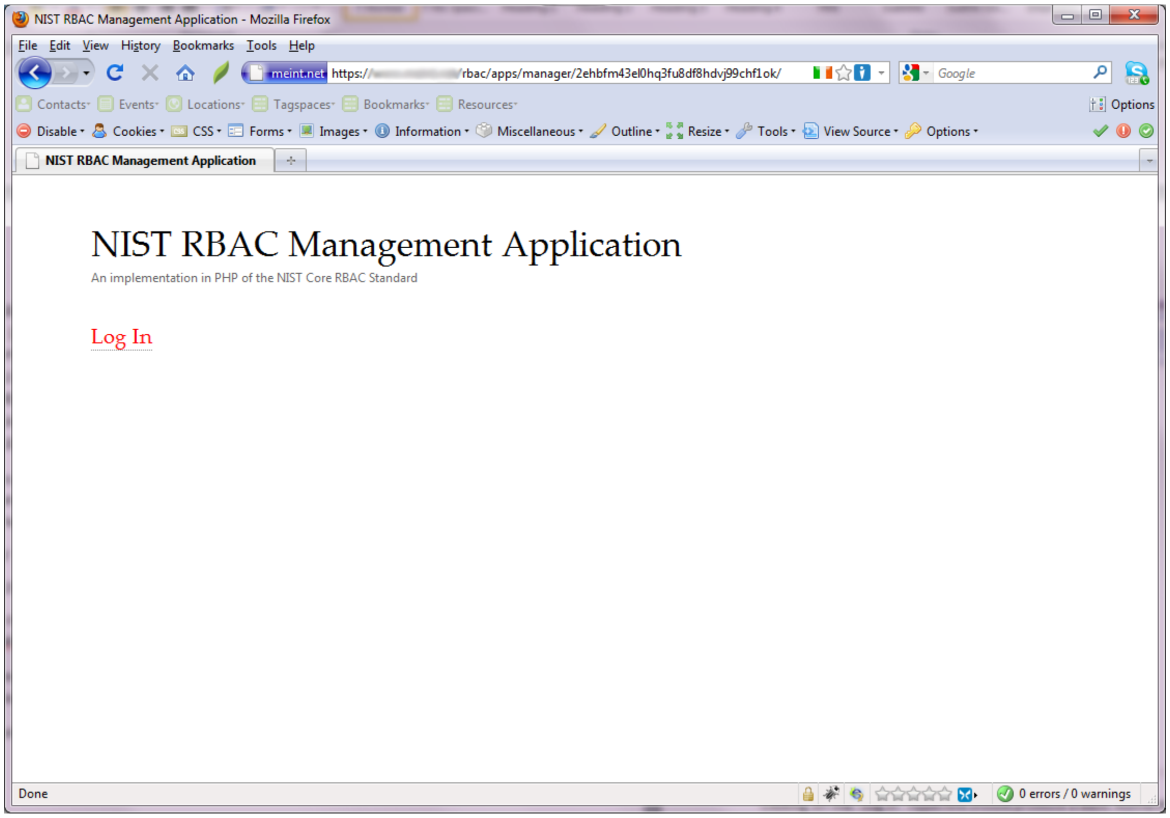
```
defined('PATH') or define('PATH', '/apps/manager');
```

(The other items of the configuration.php file have been left out because they do not directly impact runtime behaviour).

Start up your browser and invoke the url of the manager application. If you have copied over the files in the webroot of your webserver the url should be:

```
http://<fill in website url>/apps/manager/
```

You should be confronted with a page looking like this:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net
Google
Contacts Events Locations Tagspaces Bookmarks Resources
Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
NIST RBAC Management Application
Done
0 errors / 0 warnings

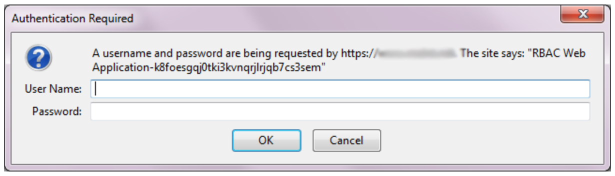
https://[redacted]/rbac/apps/manager/2ehbfm43el0hq3fu8df8hdvj99chf1ok/

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Log In

Clicking on the “**Log in**” hyperlink should produce a Basic Authentication prompt looking like this:



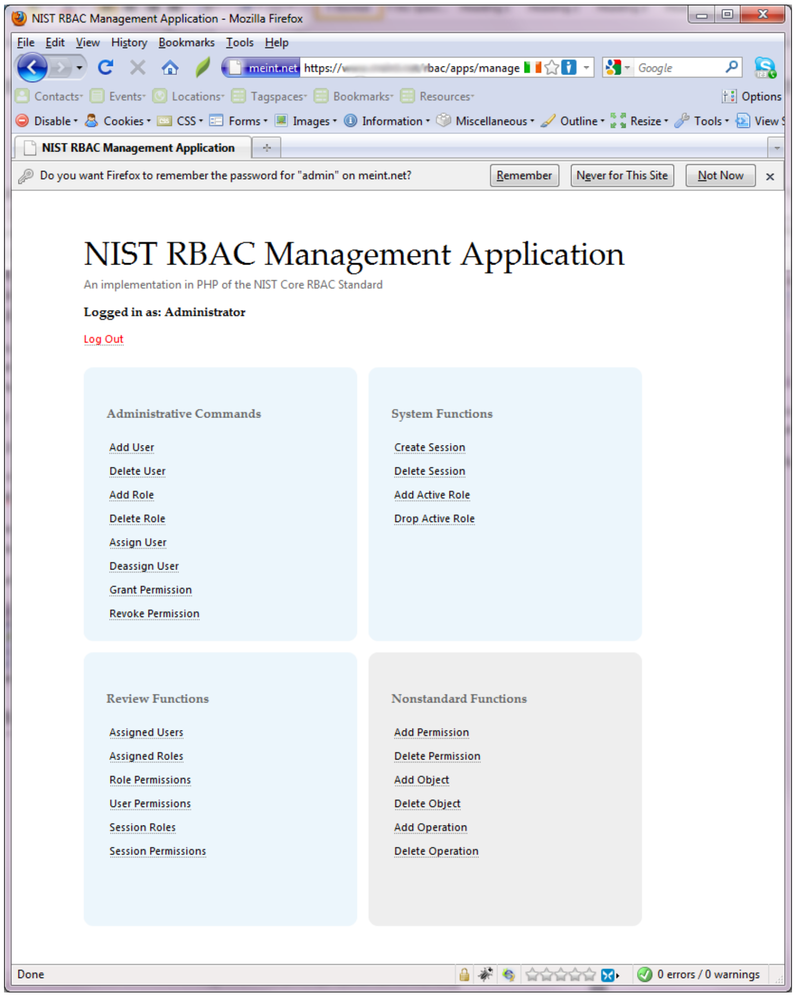
A username and password are being requested by https://[redacted]. The site says: "RBAC Web Application-k8foesgqj0tki3kvnqrjlrjqb7cs3sem"

User Name:

Password:

Both username and password are "admin". You need to replace this as quickly as you can with a different account. The following steps will show you how.

After a successful login you will see the NIST RBAC Management Application menu:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View
NIST RBAC Management Application
Remember
Never for This Site
Not Now
Done
0 errors / 0 warnings

Do you want Firefox to remember the password for "admin" on meint.net?

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Logged in as: Administrator

Log Out

Administrative Commands

- Add User
- Delete User
- Add Role
- Delete Role
- Assign User
- Deassign User
- Grant Permission
- Revoke Permission

System Functions

- Create Session
- Delete Session
- Add Active Role
- Drop Active Role

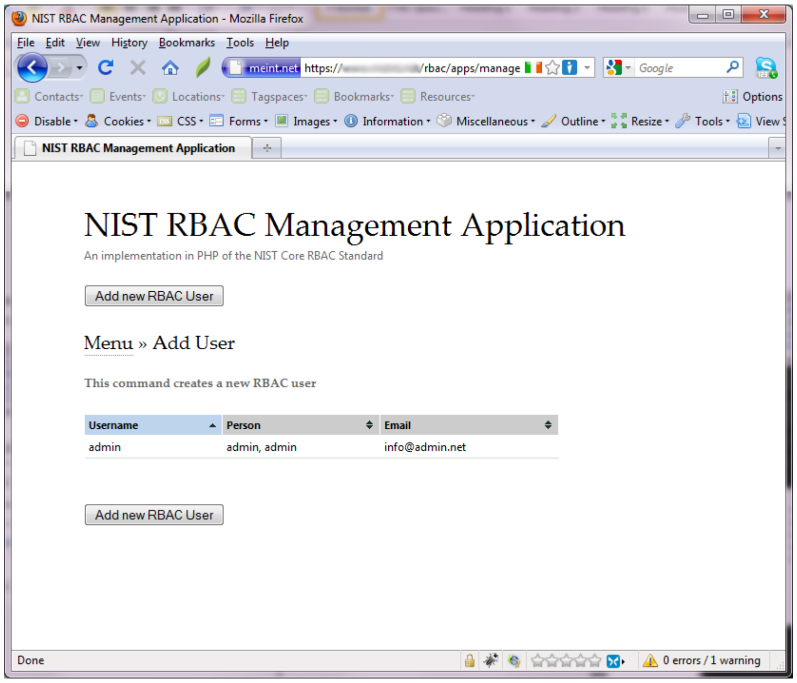
Review Functions

- Assigned Users
- Assigned Roles
- Role Permissions
- User Permissions
- Session Roles
- Session Permissions

Nonstandard Functions

- Add Permission
- Delete Permission
- Add Object
- Delete Object
- Add Operation
- Delete Operation

The first action you need to do is to click on the **Add User** function. This will add a new user to replace the default admin user.


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://[redacted]/rbac/apps/manage
NIST RBAC Management Application
Done
0 errors / 1 warning

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Add new RBAC User

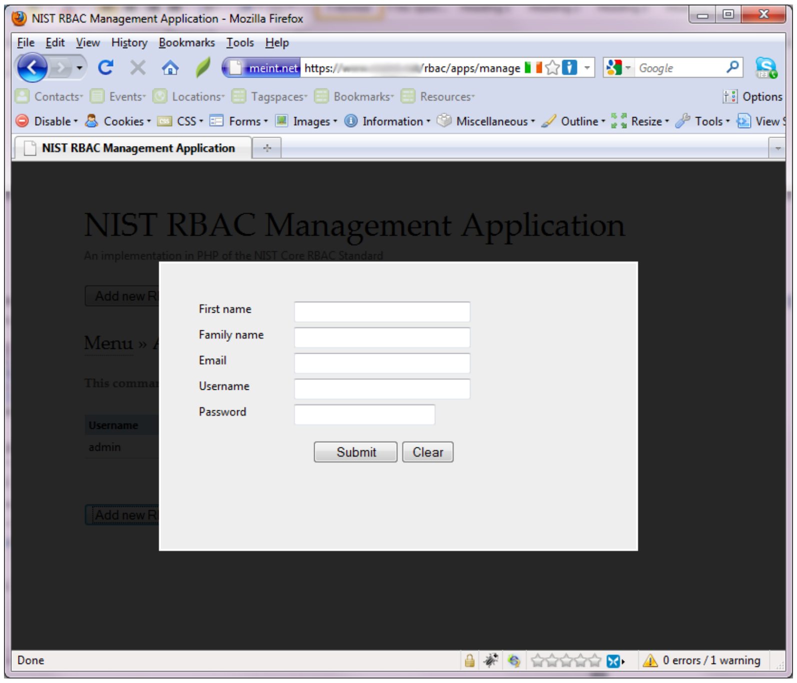
Menu » Add User

This command creates a new RBAC user

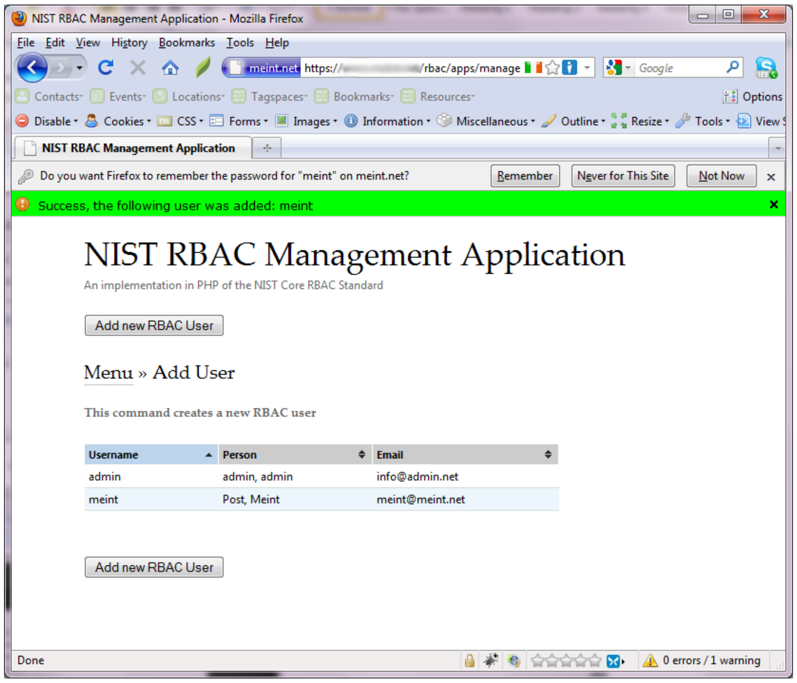
Username	Person	Email
admin	admin, admin	info@admin.net

Add new RBAC User

Click on the **Add new RBAC User** button:


NIST RBAC Management Application - Mozilla Firefox
https://[redacted]/rbac/apps/manage
First name
Family name
Email
Username
Password
Submit
Clear
Done
0 errors / 1 warning

Please fill in all the required information and click the **Submit** button. If everything went ok you should see the following screen:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https://[redacted]/rbac/apps/manage
Google
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View
NIST RBAC Management Application
Remember
Never for This Site
Not Now
Add new RBAC User
Done
0 errors / 1 warning

Do you want Firefox to remember the password for "meint" on meint.net?

Success, the following user was added: meint

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

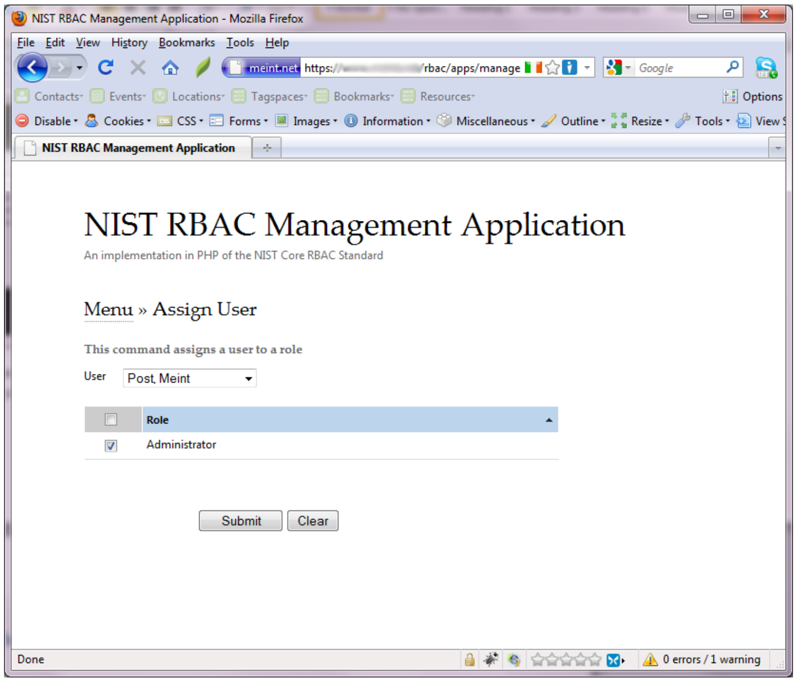
Menu » Add User

This command creates a new RBAC user

Username	Person	Email
admin	admin, admin	info@admin.net
meint	Post, Meint	meint@meint.net

You have now added an extra user. The next step is to assign the Administrator role to this new user. Click on the **Menu** hyperlink to return to the main menu. In the menu please click on the **Assign User** function, you can find it in the Administrative Commands section of the menu.

The **Assign User** function shows an overview of which roles are assigned to which user. You can use this function to assign roles to users. Select the newly added user from the dropdown and tick the checkbox next to Administrator:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https://[redacted]/rbac/apps/manage
Google
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View
NIST RBAC Management Application
Submit
Clear
Done
0 errors / 1 warning

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

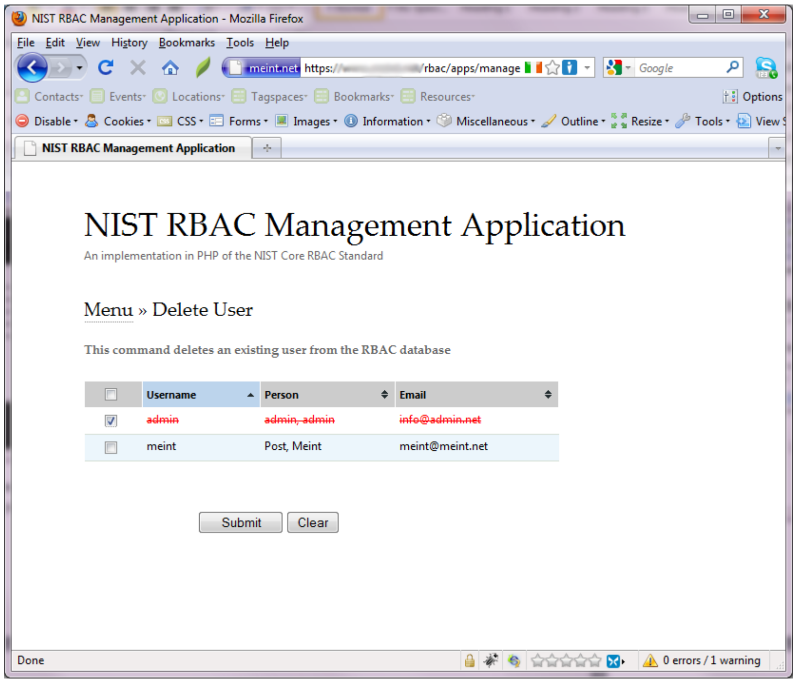
Menu » Assign User

This command assigns a user to a role

User Post, Meint

	Role
✓	Administrator

After that click on the **Submit** button. You will see a notification that the role has been succesfully assigned to your new user. Now you can delete the admin user from your system. Go back to the main menu and select **Delete User** function from the Administrative Commands section. Tick the checkbox next to the admin user, you will see the effect as shown below:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https:// /rbac/apps/manage
Google
NIST RBAC Management Application
Submit
Clear
Done
0 errors / 1 warning

NIST RBAC Management Application

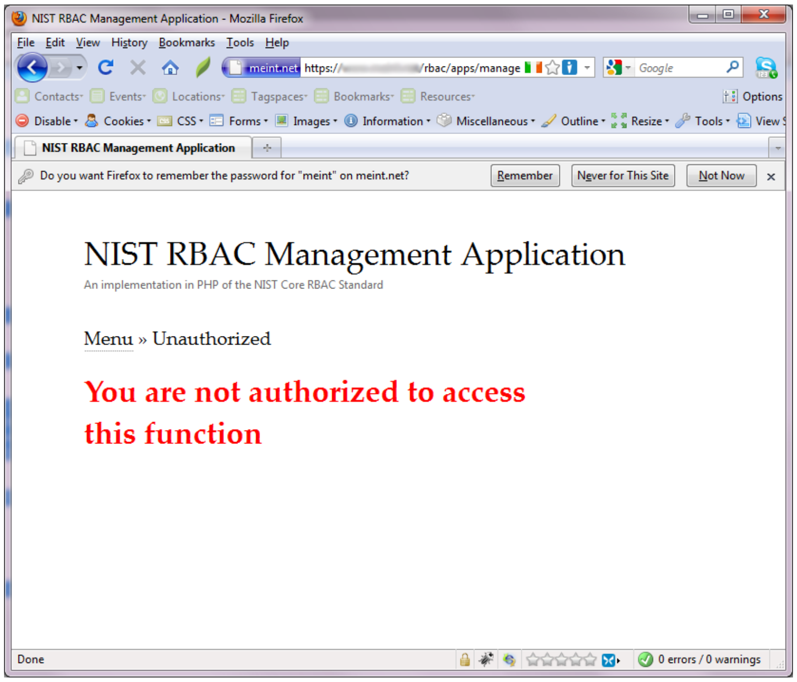
An implementation in PHP of the NIST Core RBAC Standard

Menu » Delete User

This command deletes an existing user from the RBAC database

	Username	Person	Email
✓	~~admin~~	~~admin, admin~~	~~info@admin.net~~
	meint	Post, Meint	meint@meint.net

Click on the **Submit** button to commit the change. Don't be alarmed by what happens next, you will see the following:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https:// /rbac/apps/manage
Google
NIST RBAC Management Application
Remember
Never for This Site
Not Now
Done
0 errors / 0 warnings

Do you want Firefox to remember the password for "meint" on meint.net?

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Menu » Unauthorized

You are not authorized to access this function

Because you have deleted the active admin user your current session context is destroyed and you no longer have authorisations in the Manager application. Click on the **Menu** link, this will prompt you with a Basic Authentication login dialogue. Please provide the credentials of the new account you created and click on **Ok**. You should be back in the menu again, only this time with your new account.

Congratulations on setting up the NIST RBAC PHP API Management Application!

Chapter 3: Using the NIST RBAC PHP API

So you've done all the stuff I asked in Chapter 2 and you're wondering what this all leads to. Up until now you still haven't secured your application haven't you? This chapter is about to remedy that. To help in explaining the usage of the NIST RBAC PHP API library I have added a small demo application. The demo application is a planning application for a Moving company. It contains a number of mock functions that need to be accessed by various staff members. The functions need to be bundled into roles and the roles need to be assigned to users.

The demo application can be found in folder /apps/demo. There is a configuration.php file in this folder. Open the configuration.php file with an editor. The file contains the following entries:

- PATH: [*url*] The abs path of the manager application where the leftmost slash (/) is the webroot
- INCLUDE_PATH: [*url*] The abs path of the include files (template, css, js)
- LANGUAGE: [*language code*] The name of the language file to load
- STATUS: [DEVELOPMENT/PRODUCTION] Status of the environment, determines error logging and reporting
- ERROR_LOG_PATH: [*file path*] Where the error log is written
- TIMEZONE: [*UNIX timezone*] Fill in the timezone information (for logging purposes)
- TITLE: [*alphanumeric*] This is the title of the page
- SUBTITLE: [*alphanumeric*] This is the subtitle of the page

Change the values according to your own configuration. If you have placed all the files in the webroot of your webserver and you have kept the folder names identical to the distribution and you're hosting on Unix you should have the following configuration.php contents:

```
defined('PATH') or define('PATH', '/apps/demo');
```

The other items of the configuration.php file have been left out because they do not directly impact runtime behaviour.

If you were to run the demo application you would not be able to access any function in the demo application because no authorisations have been assigned to anyone. The way that the library works is that you have to define functions that you wish to protect, determine the access levels for these functions, combine these items into permissions, bundle permissions into roles and add roles to users.

Step 1: Inspect the code

In the demo folder you will find a file named view.php. Please open this file in your editor. In this file you will find a number of functions. The available functions are:

- AddMove: Add a move
- UpdateMove: Update a move

- DeleteMove: Delete a move
- ShowMove: Show a single move
- ShowMoves: Show all moves

Each function is protected by the CheckAccess API call like below:

```
if (CheckAccess($session, 'AddMove', 'create_read')) {

      …

} else {

      notAuthorized();

}
```

The CheckAccess call is the only call you need to use in your application. It takes three arguments: a session reference, the name of the function that you want to authorize and the access rights you require for allowing access. The access rights are predefined (but can be changed if you so desire through the AddOperation/DeleteOperation functions) and are as follows:

- none
- create
- read
- update
- delete
- create_read
- create_update
- create_delete
- create_read_update
- create_read_delete
- create_update_delete
- create_read_update_delete
- read_update
- read_delete
- read_update_delete
- update_delete

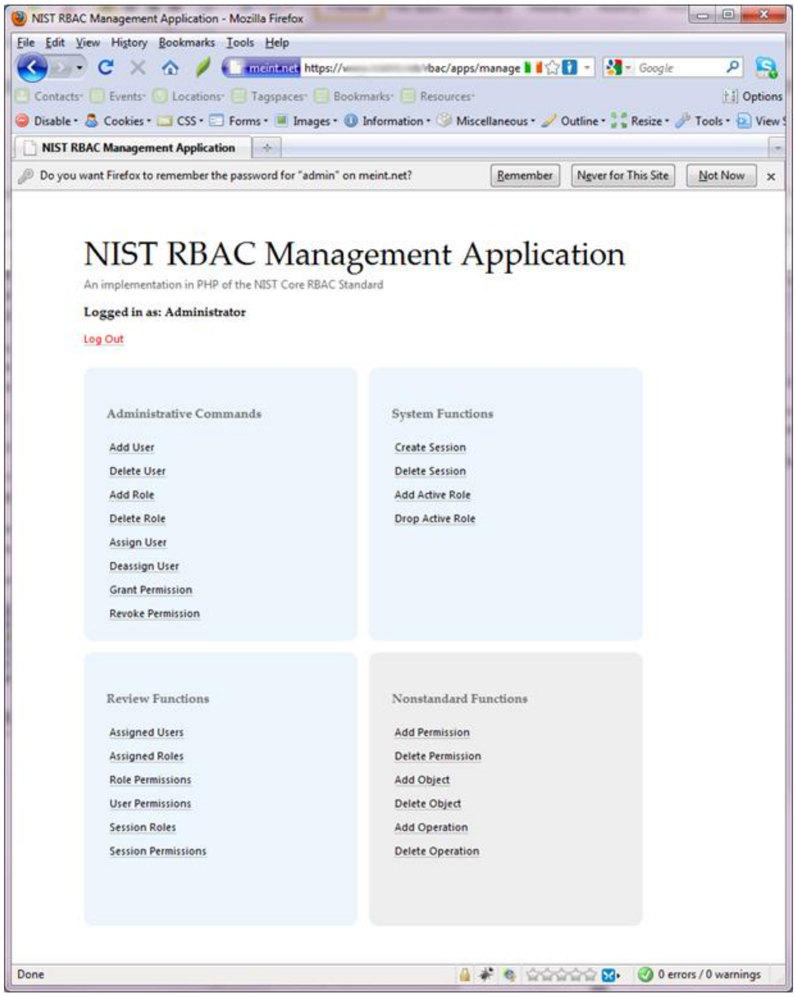
Basically there are four operations (create, read, update and delete) and they can be either on or off leading to 4^2 possibilities = 16 combinations.

In the example above the protected function is AddMove and the required access rights are “create_read”. It is up to the protected function to make sure these access rights are in line with the working of the function, i.e. typically a create_read function should allow SQL SELECT and INSERT statements but not UPDATE or DELETE statements.

Step 2: Add functions to the RBAC data model

In its current state the demo application will not function because the application functions have not been added to the RBAC data model. This needs to be done before the functions can be mapped to roles.

For the demo tutorial we will be using the NIST RBAC Management Application to add functions, create roles, add users and assign users. In Chapter 3 you have activated the Management Application, please use the url from Chapter 3, step 4, to access the Management Application. You should see the standard menu interface:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
NIST RBAC Management Application
Remember
Never for This Site
Not Now
Done
0 errors / 0 warnings

Do you want Firefox to remember the password for "admin" on meint.net?

NIST RBAC Management Application

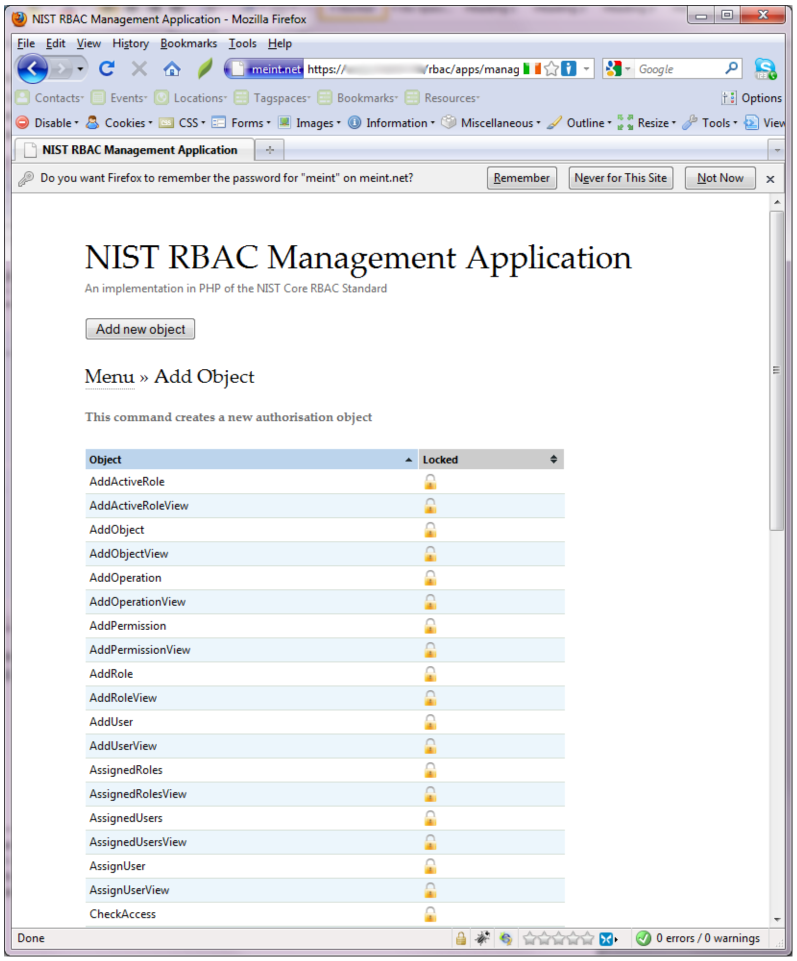
An implementation in PHP of the NIST Core RBAC Standard

Logged in as: Administrator

Log Out

From the menu interface select function **Add Object**, you can find it in the Nonstandard Functions section[2]. This will open up the Add Object interface as shown below:

[2] The NIST RBAC standard presupposes an already existing administration of objects, operations and operations. Hence the standard doesn't supply functions for managing these items. Since this will not be the case for most applications using this API implementation I have added management functions for these items. As they are not part of the formal standard I have named them "Nonstandard Functions".


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Do you want Firefox to remember the password for "meint" on meint.net?
Remember
Never for This Site
Not Now
Done
0 errors / 0 warnings

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

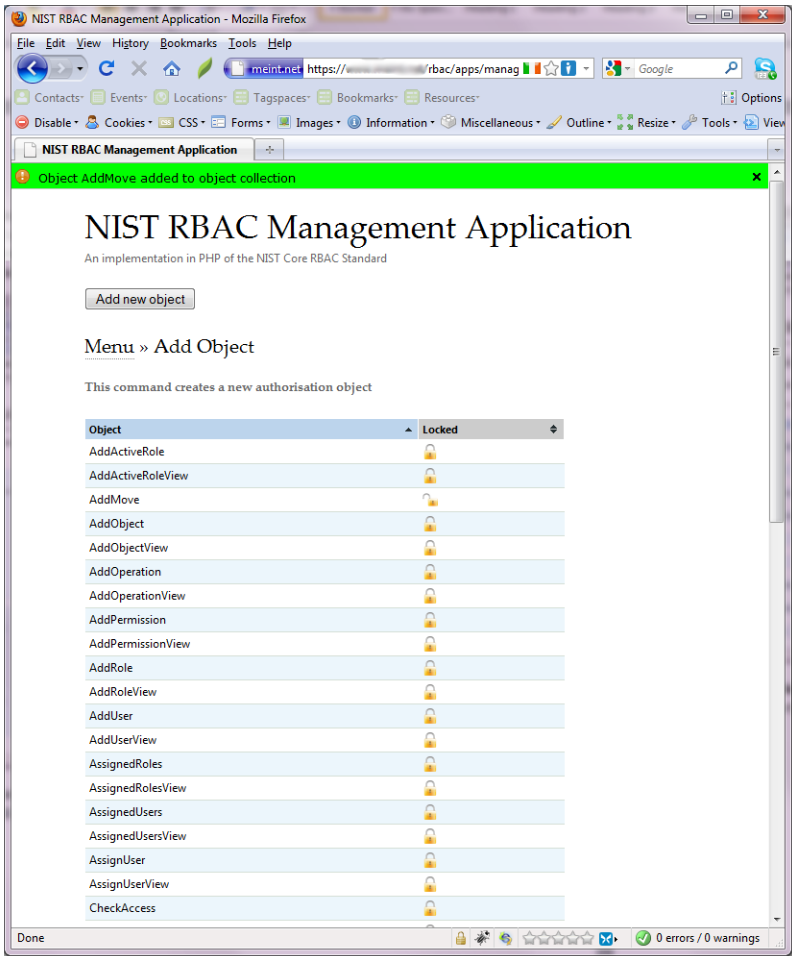
Add new object

Menu » Add Object

This command creates a new authorisation object

Object	Locked
AddActiveRole	🔒
AddActiveRoleView	🔒
AddObject	🔒
AddObjectView	🔒
AddOperation	🔒
AddOperationView	🔒
AddPermission	🔒
AddPermissionView	🔒
AddRole	🔒
AddRoleView	🔒
AddUser	🔒
AddUserView	🔒
AssignedRoles	🔒
AssignedRolesView	🔒
AssignedUsers	🔒
AssignedUsersView	🔒
AssignUser	🔒
AssignUserView	🔒
CheckAccess	🔒

Click on the **Add new object** button. A modal window will pop up that allows entry of an **Object name**. Please type in: AddMove and click on **Submit**. You should see the following screen:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https://[redacted]/rbac/apps/manag
Google
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View
NIST RBAC Management Application
Done
0 errors / 0 warnings

Object AddMove added to object collection

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Add new object

Menu » Add Object

This command creates a new authorisation object

Object	Locked
AddActiveRole	🔒
AddActiveRoleView	🔒
AddMove	🔓
AddObject	🔒
AddObjectView	🔒
AddOperation	🔒
AddOperationView	🔒
AddPermission	🔒
AddPermissionView	🔒
AddRole	🔒
AddRoleView	🔒
AddUser	🔒
AddUserView	🔒
AssignedRoles	🔒
AssignedRolesView	🔒
AssignedUsers	🔒
AssignedUsersView	🔒
AssignUser	🔒
AssignUserView	🔒
CheckAccess	🔒

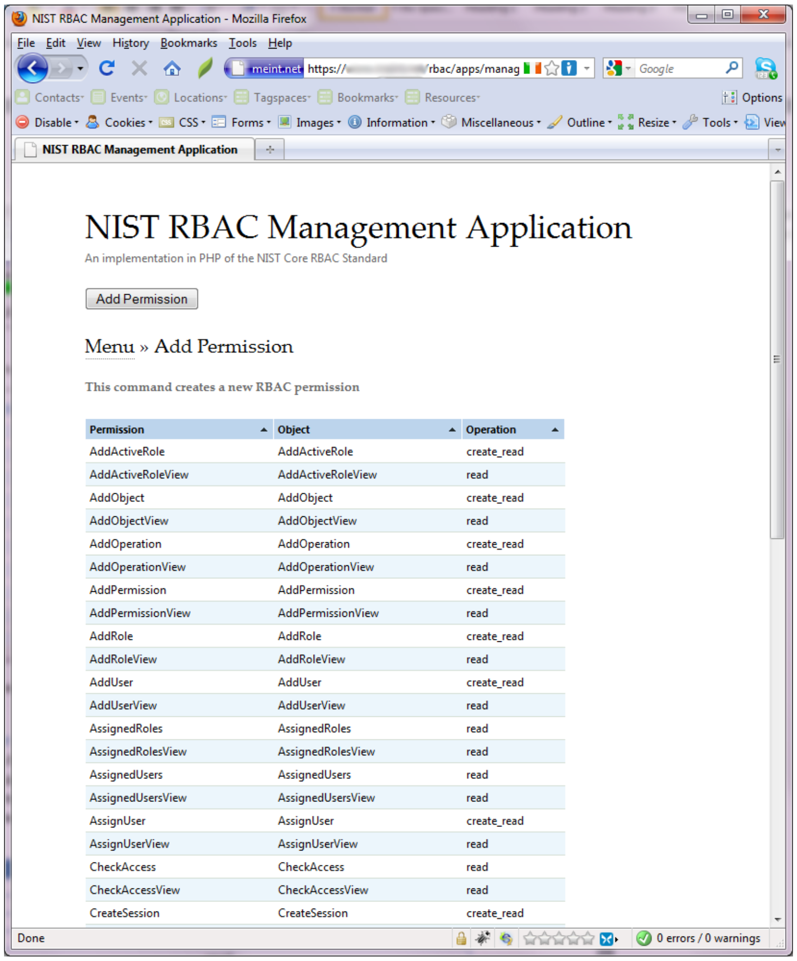
Repeat this procedure for the other demo functions as well: UpdateMove, DeleteMove, ShowMove and ShowMoves.

Step 3: Create permissions

According to the NIST RBAC standard permissions consist of objects and operations. In our setup objects are user definable and operations are predefined (but can be changed if desired). The step into creating permissions therefore exists in combining the objects that we have added in step 2 with the predefined operations. Looking at the code in view.php we can verify the following combinations:

Object	Operation
AddMove	create_read
UpdateMove	read_update
DeleteMove	read_delete
ShowMove	read
ShowMoves	read

Go back to the menu and select menu function **Add Permission** from Nonstandard functions. You will see this screen:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https://[redacted]rbac/apps/manag
Google
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View
NIST RBAC Management Application
Done
0 errors / 0 warnings

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Add Permission

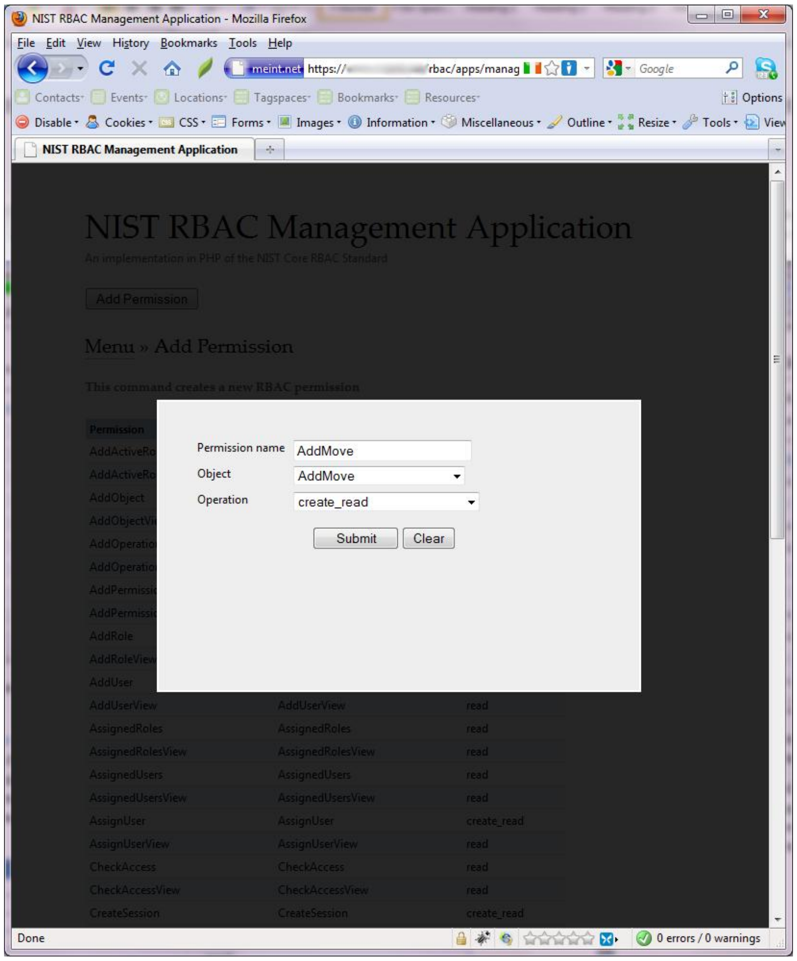
Menu » Add Permission

This command creates a new RBAC permission

Permission	Object	Operation
AddActiveRole	AddActiveRole	create_read
AddActiveRoleView	AddActiveRoleView	read
AddObject	AddObject	create_read
AddObjectView	AddObjectView	read
AddOperation	AddOperation	create_read
AddOperationView	AddOperationView	read
AddPermission	AddPermission	create_read
AddPermissionView	AddPermissionView	read
AddRole	AddRole	create_read
AddRoleView	AddRoleView	read
AddUser	AddUser	create_read
AddUserView	AddUserView	read
AssignedRoles	AssignedRoles	read
AssignedRolesView	AssignedRolesView	read
AssignedUsers	AssignedUsers	read
AssignedUsersView	AssignedUsersView	read
AssignUser	AssignUser	create_read
AssignUserView	AssignUserView	read
CheckAccess	CheckAccess	read
CheckAccessView	CheckAccessView	read
CreateSession	CreateSession	create_read

The permissions shown on the screen are the permissions for the NIST RBAC Management Application itself.

Now you need to create your own permissions. Click on **Add Permission** to start this process. A modal window will appear allowing you to enter three items: a permission name, permission object and permission operation. The latter two are prefilled drop down lists containing available objects and operations. To give you an example for the AddMove permission see below:


NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Menu » Add Permission

This command creates a new RBAC permission

Permission name: AddMove
Object: AddMove
Operation: create_read

Permission		
AddActiveRo		
AddActiveRo		
AddObject		
AddObjectVi		
AddOperatio		
AddOperatio		
AddPermissi		
AddPermissi		
AddRole		
AddRoleView		
AddUser		
AddUserView	AddUserView	read
AssignedRoles	AssignedRoles	read
AssignedRolesView	AssignedRolesView	read
AssignedUsers	AssignedUsers	read
AssignedUsersView	AssignedUsersView	read
AssignUser	AssignUser	create_read
AssignUserView	AssignUserView	read
CheckAccess	CheckAccess	read
CheckAccessView	CheckAccessView	read
CreateSession	CreateSession	create_read

You do not need to give the permission the same name as the object, it can be something altogether differently as long as it makes sense to you.

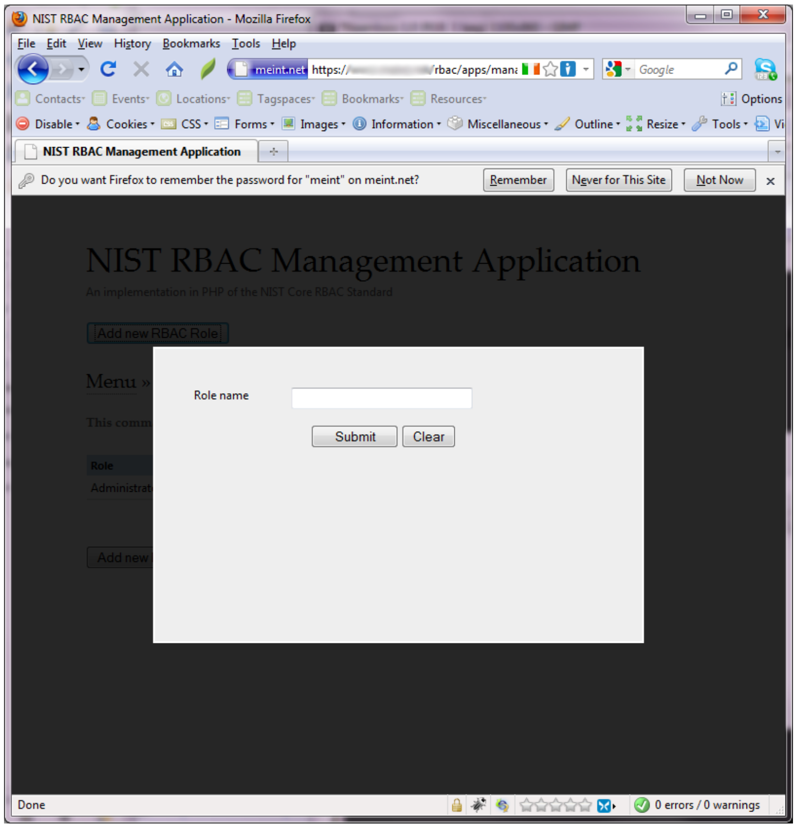
Click on **Submit** after you have supplied the correct information. Repeat the process for the other functions and operations as outlined in the table above.

Step 4: Create roles

Permissions need to be aggregated into roles to provide business functionality. For the demo application we will define four roles: Manager, Coordinator, Mover and Customer. Each role has one or more permissions like follows:

	AddMove	UpdateMove	DeleteMove	ShowMove	ShowMoves
Manager					X
Coordinator	X	X			X
Mover			X		X
Customer				X	

To add permissions to role you need to go back to the main menu and select the function **Add Role**. In the Add Role screen please click on the **Add new RBAC Role** button. This will show the following modal window:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https:// /rbac/apps/man
Google
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools
NIST RBAC Management Application
Remember
Never for This Site
Not Now
Add new RBAC Role
Menu »
Role
Administrat
Add new
Submit
Clear
Done
0 errors / 0 warnings

Do you want Firefox to remember the password for "meint" on meint.net?

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

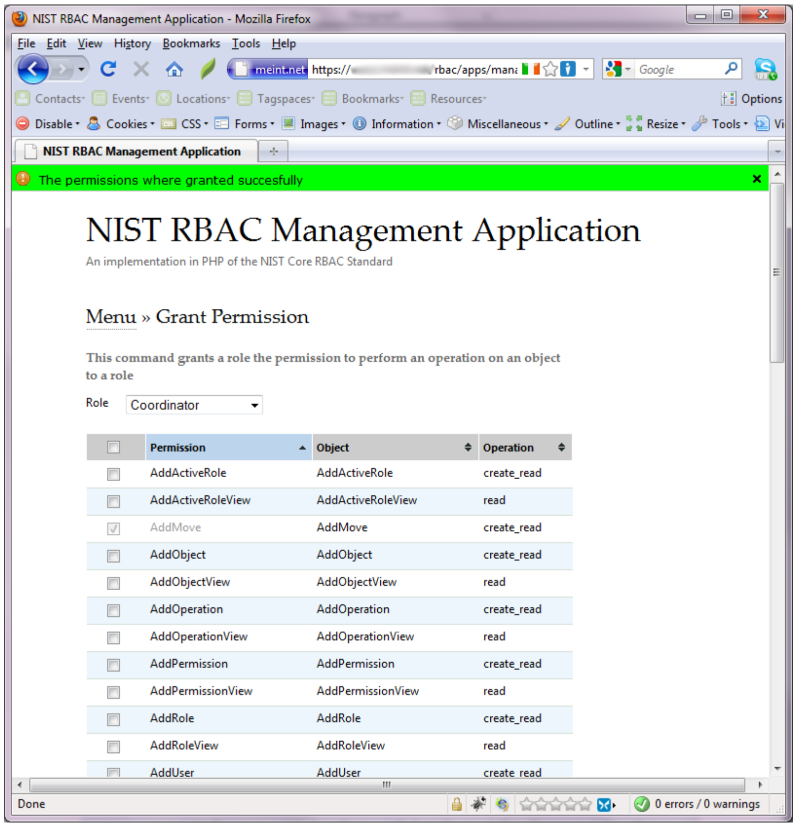
Role name

Please type in: Manager and click on **Submit**. Repeat the action for Coordinator, Mover and Customer.

Step 5: Add permissions to roles

Now it's time to add some substance to it all by adding permissions to roles. Go to the main menu and select function **Grant Permission**. In the Grant Permission screen select the role Manager from the dropdown menu and tick the checkbox for permission **ShowMoves**. Click on **Submit** to execute the change. You will see the Manager role selected in the updated screen and you will also see that the ShowMoves permission is now greyed out because it has been integrated with the Manager role.

Repeat the same actions for Coordinator (with permissions AddMove, UpdateMove and ShowMoves), Mover (with permissions DeleteMove and ShowMoves) and Customer (with permission ShowMove).


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools
NIST RBAC Management Application
Done
0 errors / 0 warnings

The permissions where granted succesfully

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

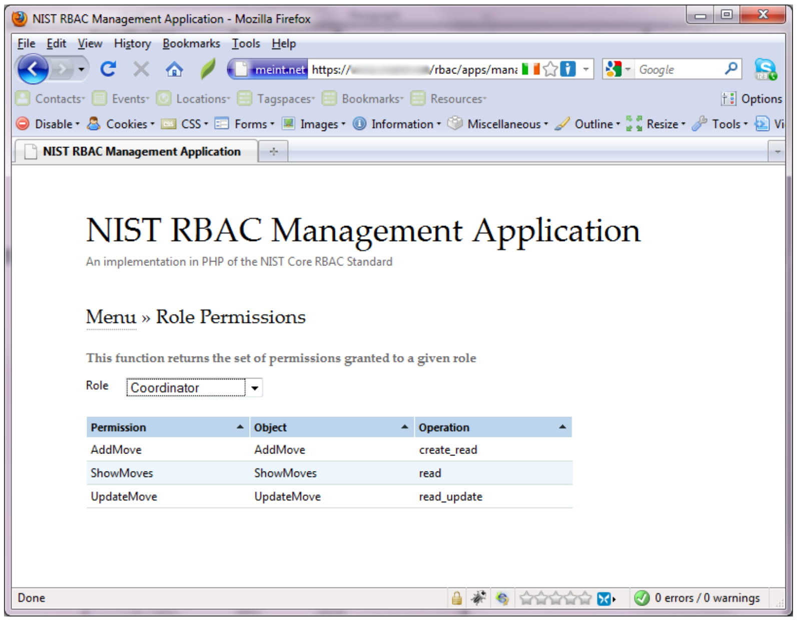
Menu » Grant Permission

This command grants a role the permission to perform an operation on an object to a role

Role: Coordinator

	Permission	Object	Operation
☐	AddActiveRole	AddActiveRole	create_read
☐	AddActiveRoleView	AddActiveRoleView	read
✓	AddMove	AddMove	create_read
☐	AddObject	AddObject	create_read
☐	AddObjectView	AddObjectView	read
☐	AddOperation	AddOperation	create_read
☐	AddOperationView	AddOperationView	read
☐	AddPermission	AddPermission	create_read
☐	AddPermissionView	AddPermissionView	read
☐	AddRole	AddRole	create_read
☐	AddRoleView	AddRoleView	read
☐	AddUser	AddUser	create_read

You can verify whether you have assigned the permissions correctly to each role by going back to the main menu and select the function **Role Permissions** from section Review Functions. In the Role Permissions screen select the role you want to review from the dropdown list. For example select Coordinator and you should see the following:


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools
NIST RBAC Management Application
Done
0 errors / 0 warnings

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Menu » Role Permissions

This function returns the set of permissions granted to a given role

Role: Coordinator

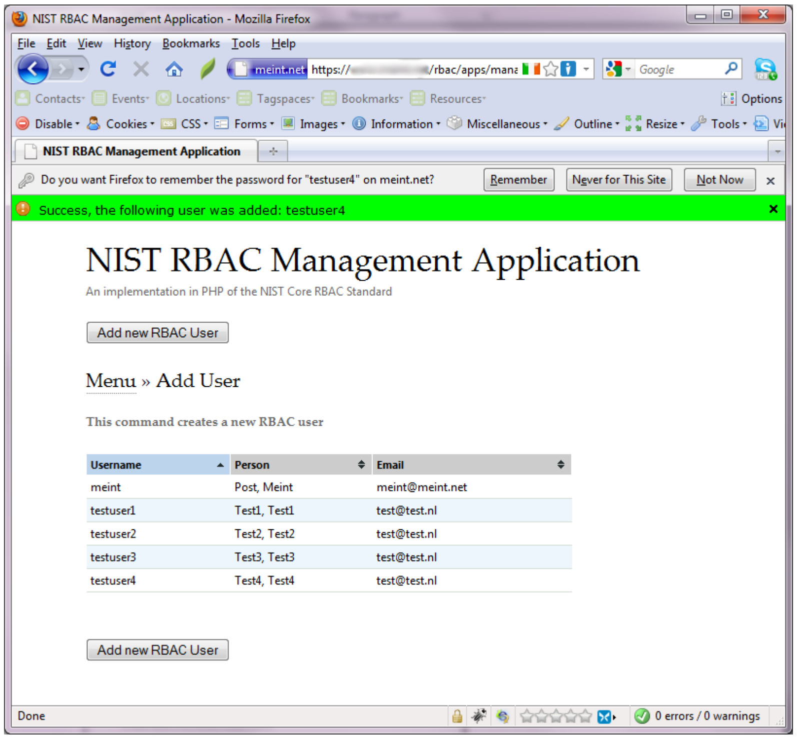
Permission	Object	Operation
AddMove	AddMove	create_read
ShowMoves	ShowMoves	read
UpdateMove	UpdateMove	read_update

As you can see this combines the information from the table in Step 3 with the information in the table of Step 4.

Step 6: Assign users to roles

We're nearing the end of this small tutorial, only one step to take before executing the demo: assigning users to roles.

First create four test users that we can each assign one of the roles we created in the previous steps. Go to the main menu and click on **Add User**. For detailed instructions see Chapter2, Step 4. Fill in whatever you like in the Add User modal window but use as the username testuser1 and use password "test". Repeat this with testuser2, testuser3 and testuser4 as usernames and "test" as the password. See the example below for the desired result.


NIST RBAC Management Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Do you want Firefox to remember the password for "testuser4" on meint.net?
Remember
Never for This Site
Not Now

Success, the following user was added: testuser4

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Add new RBAC User

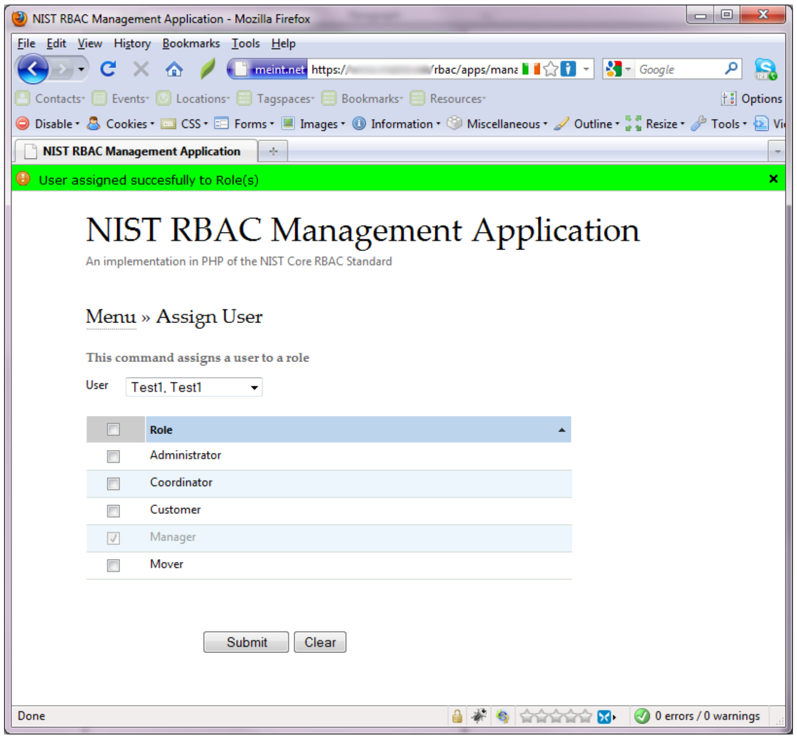
Menu » Add User

This command creates a new RBAC user

Username	Person	Email
meint	Post, Meint	meint@meint.net
testuser1	Test1, Test1	test@test.nl
testuser2	Test2, Test2	test@test.nl
testuser3	Test3, Test3	test@test.nl
testuser4	Test4, Test4	test@test.nl

Add new RBAC User

Go to the main menu and select function **Assign User**. In the Assign User screen select the first test user, tick the checkbox next to the Manager role and click on **Submit**. You will see the following result:


User assigned succesfully to Role(s)

NIST RBAC Management Application

An implementation in PHP of the NIST Core RBAC Standard

Menu » Assign User

This command assigns a user to a role

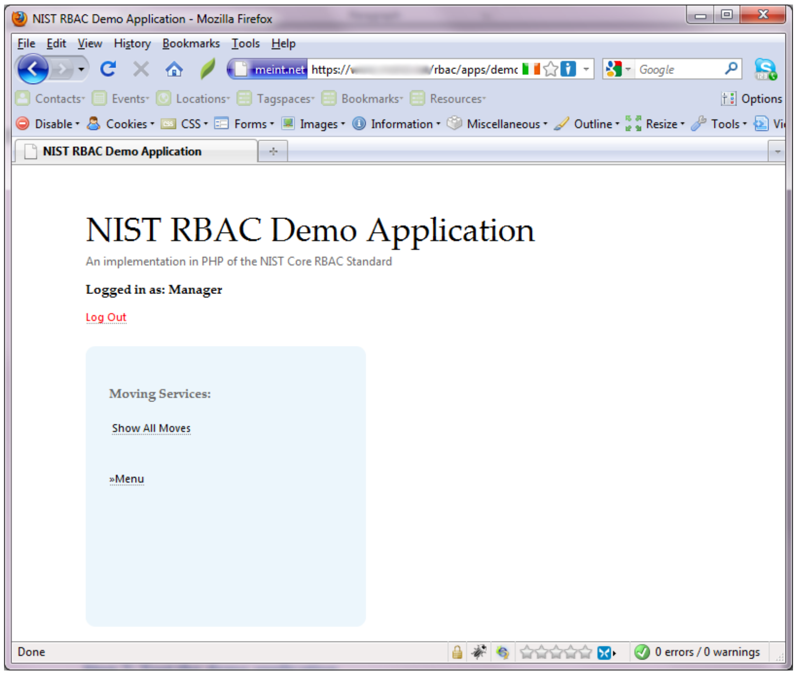
User Test1, Test1

	Role
☐	Administrator
☐	Coordinator
☐	Customer
✓	Manager
☐	Mover

Repeat this for the second test user and assign the Coordinator role, the third test user assign the Mover role and the fourth test user assign the Customer role.

Step 7: Test the demo application

Now it's time for the last step, running the demo application and see our work in action. Navigate to the demo application, http://<www.yourserver.org>/apps/demo and click on the **Log in** hyperlink. In the Basic Authentication dialog box provide as username testuser1 and as password test.


NIST RBAC Demo Application

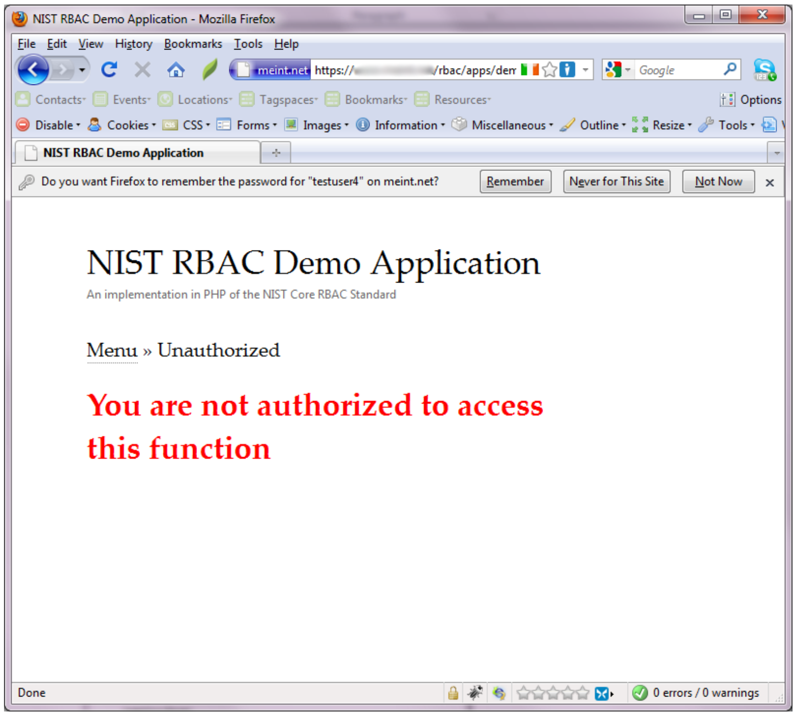
An implementation in PHP of the NIST Core RBAC Standard

Logged in as: Manager

The personalised menu for the Mover application should come up identifying you as role Manager and having access to one function: ShowMoves.

You can verify all other roles by logging out and logging in with different usernames.

You can also verify that you can't access functions you're not authorized for by hacking the url. If you're logged in as Customer try changing the url by changing the action query string parameter into /?action=UpdateMove. You should be seeing a warning screen with unauthorized access.


NIST RBAC Demo Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
meint.net https:// /rbac/apps/dem
Google
Contacts Events Locations Tagspaces Bookmarks Resources Options
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools
NIST RBAC Demo Application
Remember
Never for This Site
Not Now
Done
0 errors / 0 warnings

Do you want Firefox to remember the password for "testuser4" on meint.net?

NIST RBAC Demo Application

An implementation in PHP of the NIST Core RBAC Standard

Menu » Unauthorized

You are not authorized to access this function

This concludes the tutorial.