



## CubiAccessControl

describe how Cubi implement access control

Phase-Implementation

Updated Jun 16, 2012 by [agus.suhartono](#)

## Cubi Access Control

Cubi provides various implementations to control users access of application resources.

### User authentication

Cubi uses authentication service (modules/service/authService.php) to authenticate user by given username and password.

Current authentication service authenticates user by checking Cubi user table. Such logic can be modified to customer environment. For example, user can be authenticated through LDAP server in this service. Such customization can be implemented by overriding Cubi "authService".

### General page (view) access control ¶

Access service can be used for simple page level access control. AccessService has a configuration file (accessService.xml) that defines the view access permission for certain roles which can be retrieved in user profile service. Please see an example below.

```
<?xml version="1.0" standalone="no"?>
<PluginService Name="accessService" Class="accessService">
  <access-constraint>
    <view-collection>
      <view name="shared.CalendarView">
        <role name="admin"/>
        <role name="member"/>
      </view>
      <view name="demo*"> <!-- regular expression in the view name -->
        <role name="admin"/>
        <role name="member"/>
      </view>
    </view-collection>
  </access-constraint>
</PluginService>
```

The xml configuration file is easy to understand. Customer needs to put their own logic in the accessService.xml.

### Role-based access control (RBAC)

The basic ideas of Cubi role-based access control is to define how a role can operate application resource. When defining a RBAC model, the following conventions are useful:

- User - A person or automated agent. A user can have multiple roles.
- Role - Job function or title which defines an authority level. A role can be assigned to multiple users. A role can have multiple permissions.
- Resource - An object that has certain logic in an application.
- Action - An operation that can change the status of a resource
- Permissions - An approval of access to a resource. It defines how a role executes actions of a resource. A permission can be assigned to multiple roles.

### Define resource and its actions

Each Cubi module has its mod.xml under the module root directory. In mod.xml, there is a "ACL" section which can have multiple resources. Each resource can have more than 1 actions. For example,

```
<ACL>
  <Resource Name="User">
    <Action Name="Administer_Users" Description="Administration of users"/>
  </Resource>
</ACL>
```

### Link Access resource action to Object

Each Openbiz object can have an "Access" attribute with format of "resource.action". For example,


```
<EasyView Name="UserListView"... Access="User.Administer_User">
```

sets the administer user access to the system.view.UserListView. In another work, only roles whose are granted with User.Administer\_User can access UserListView.

### Assign Role permission to resource action

In the role management view, user can pick "Allow" or "Deny" to all available resource actions. Say, we give role "member" a "Deny" to

User.Administer\_User. Then when a user with member role tries to access the RoleListView, an access deny page will shown to the user.





### Details of role

Detailed attributes of a given role

**Name** Member

**Description** General registered users

**Status** 



### Role permissions

Manage the permissions of the given role by setting access level to resource actions

Action Id	Module	Resource	Action	Access Level
1	system	User	Administer_Users	<input type="text" value=""/>
2	system	User	Administer_User_ACL	<input type="text" value=""/>
3	system	Role	Administer_Roles	<input type="text" value=""/>
4	system	Module	Administer_Modules	<input type="text" value=""/>
13	eventlog	EventLog	Administer_EventLog	<input type="text" value=""/>
12	eventlog	EventLog	Access_EventLog	<input type="text" value=""/>
7	menu	Menu	Administer_Menu	<input type="text" value=""/>
8	email	EmailQueue	Administer_Email_Queue	<input type="text" value=""/>
9	email	EmailLog	Administer_Email_Log	<input type="text" value=""/>
10	help	Help	Administer_Help	<input type="text" value=""/>

Access attribute can be given to View, Form, Element, DataObj, Menu item.

## Group-based visibility control

Role is used to control if an action on a resource can be conducted by a user, while there are many cases that want different users see different data set. For example, sales data should be viewable to not only all sales, but also finance people and marketing people. This is so-called data "visibility".

### Explicit Group

Cubi uses "group" to control data visibility. In order to add visibility control on certain data, a new field "group\_id" can be added into the corresponding data object. Then custom logic can be added in DataObject AccessRule. For example,

- To set data visible to its group only, you can set

```
<BizDataObj Name="SalesDO" AccessRule="{tx}@vis:group(group_id){/tx}" ...>
```

- To set data visible to its owner only, you can set

```
<BizDataObj Name="MailDO" AccessRule="{tx}@vis:group(owner_id){/tx}" ...>
```

@vis is the alias of visibility service. Its group method will return additional search rule that limit the query results.

All service alias can be defined in \$g\_ServiceAlias at app\_init.php. Once service alias is defined, Openbiz expression engine can invoke corresponding service method declared in expression string.

### User-based visibility control

Sometimes, we want to add finer visibility control that gives each user access to each data record. In this case, an intersection table is recommended to link user and data with many to many relationship. For example, if you want to give multiple users permission to access a critical report, you can create an intersection table named "report\_user" which stores report id and user id. This table is used to tell who has access to which reports.

This type of many to many relationship and user interface are implemented in many Cubi modules (e.g. user/role, user/group). Referring to existing Cubi implementation will make development much quicker.

## Individual data Access Control

In order to allow permission control at individual data record, Cubi applies Unix file permission concept on DataObject.

## Unix file permission

Here briefly introduces how Unix-like systems control permission of files.

Permissions on Unix-like systems are managed in three distinct classes. These classes are known as user, group, and others.

- Files and directories are owned by a user. The owner determines the file's owner class. Distinct permissions apply to the owner.
- Files and directories are assigned a group, which define the file's group class. Distinct permissions apply to members of the file's group members. The owner doesn't need to be a member of the file's group.
- Users who are not the owner, nor a member of the group, comprise a file's others class. Distinct permissions apply to others.

The effective permissions are determined based on the user's class. For example, the user who is the owner of the file will have the permissions given to the owner class regardless of the permissions assigned to the group class or others class.

There are three specific permissions on Unix-like systems that apply to each class:

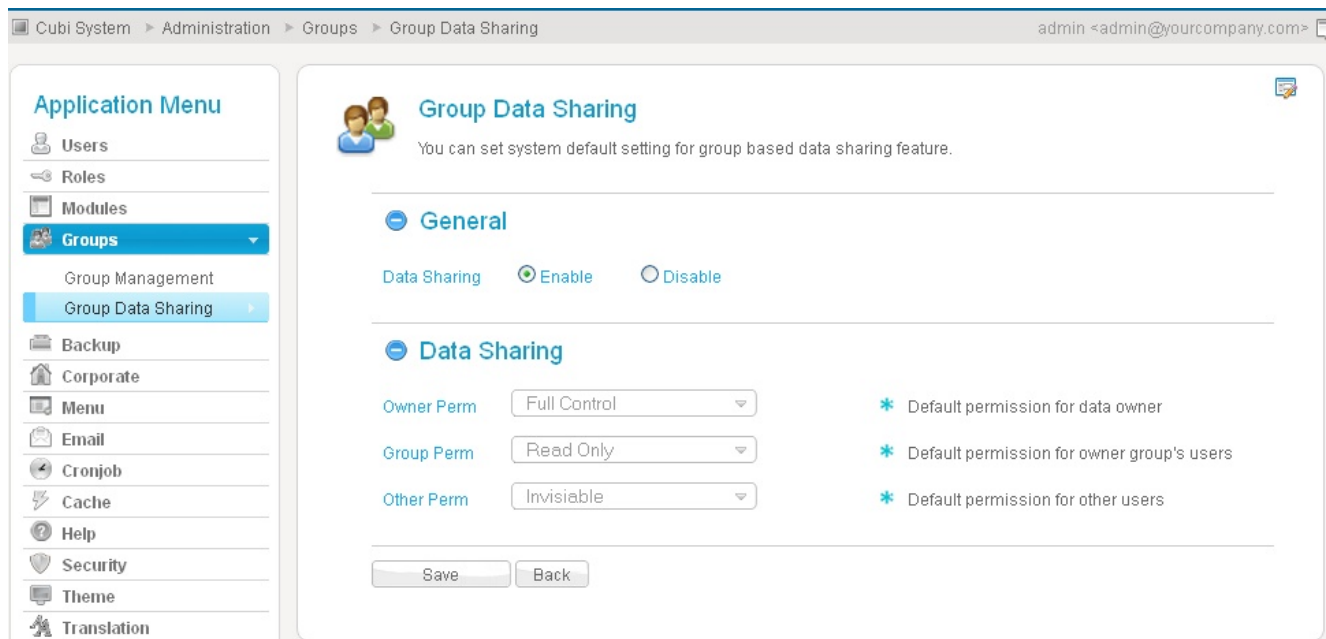
- The read permission, which grants the ability to read a file.
- The write permission, which grants the ability to modify a file.
- The execute permission, which grants the ability to execute a file.

In Unix system, a file has attributes that tells

- what the owner can do (read, write or execute)
- what the group members can do (read, write or execute)
- what other users can do (read, write or execute)

## Enable Group Data Sharing

First, group data sharing needs to be enabled to allow individual record permission control. Cubi administer can manage it on Group Data Sharing view.



## Configure DataObject

In Cubi system, each data record can be treated as a file. To apply file permission control on data record, the following fields are needed in table column.

- owner\_id
- group\_id
- group\_perm
- other\_perm

And corresponding fields should be added in DataObject.

```
<BizField Name="owner_id" Column="owner_id" ValueOnCreate="{@profile:Id}" Required="N" Type="Number"/>
<BizField Name="group_id" Column="group_id" ValueOnCreate="{@profile:default_group}" Required="N" Type="Number"/>
<BizField Name="group_perm" Column="group_perm" ValueOnCreate="{BizSystem::GetDefaultPerm(group)}" Required="N" Type="Nl
<BizField Name="other_perm" Column="other_perm" ValueOnCreate="{BizSystem::GetDefaultPerm(other)}" Required="N" Type="Nl
```

Also this DataObject needs to have an attribute DataPermControl="Y". For example,

```
<BizDataObj Name="ContactDO" DataPermControl="Y" Class="BizDataObj" DBName="Default" Table="contact" ...
```

## Configure Form

To allow modify a data record permissions, a new element should be added in the list Form of a DataObject.

```
<Element Name="fld_share" Class="ColumnShare"
  MyPrivateImg="{RESOURCE_URL}/contact/images/icon_contact.gif"
  MySharedImg="{RESOURCE_URL}/contact/images/icon_contact_shared.gif"
  MyAssignedImg="{RESOURCE_URL}/contact/images/icon_contact_assigned.gif"
  MyDistributedImg="{RESOURCE_URL}/contact/images/icon_contact_distributed.gif"
  GroupSharedImg="{RESOURCE_URL}/contact/images/icon_contact_shared_group.gif"
  OtherSharedImg="{RESOURCE_URL}/contact/images/icon_contact_shared_other.gif"
  FieldName="create_by" Label="Share" Sortable="Y" AllowURLParam="N" Translatable="N" OnEventLog="N" Link="javascript::">
  <EventHandler Name="fld_share_onclick" Event="onclick" Function="LoadDialog(common.form.DataSharingForm,{@:Elem[fld_Id]
</Element>
```

The Form look like the screenshot below with this ColumnShare element.

Share	Contact Name	Phone	Mobile	Ordering	Type
<input type="checkbox"/>	admin			0	Business
<input type="checkbox"/>	Rocky, Swen			50	Business
<input type="checkbox"/>	Jixian, Wang	+86 10 6497 9191	+86 139 1015 4220	50	Business
<input type="checkbox"/>	Wang, Ou	+86 10 64979191		50	Business
<input type="checkbox"/>	test, li			50	Business

A permission setting dialog will popup after clicking the "Share" icon. User can change the permission on this dialog.

test, li

Data Type: My Data Record

Data Creator: admin

Data Owner: admin

Owner Perm: Full Control

Group: Sales group

Group Perm: Read Only

Other Perm: Invisible

OK Cancel

► [Sign in](#) to add a comment

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)