

Penn authorization system

Over the last 6 years Penn has built and maintained our Framework for Administrative Systems and Technologies (FAST). One of the components is group management, and one of the components is privilege management. These components are local for each application, not central. We do have 60+ custom administrative applications using FAST and this security system. To integrate FAST and Grouper, we define the group locally in FAST, and define it in grouper, and when a user logs in, we can ask grouper if the user is in any of the app specific groups, if so, dynamically assign the user to the FAST group(s). Perhaps a similar setup would work to integrate FAST with Internet2 central privilege management. So this framework is not mutually exclusive with Internet2.

Im not saying Internet2 should adopt the this piece of the FAST framework (many reasons, e.g. it is oracle specific), but parts of it are interesting. Anyways, this is how it works:

The groups are hierarchical, and the privilege resources can be bundled in hierarchical groups.

This screenshot is an example of the group hierarchy. It is not hierarchical on a membership basis, but rather, for permissions. So in this case, a senior loan officer would inherit permissions from the loan office role, which inherits permissions from the staff role. But if you asked FAST if a senior loan officer user is a member of the "Staff" group, it would say "no" if there is not an explicit user/group membership assignment.

The screenshot shows the FAST Admin Console interface. On the left is a navigation menu with the FAST logo and 'Admin Console' text. The main content area is titled 'Authorization - Edit/Delete Group'. It features a search box 'Find a FAST group to edit or delete' with a dropdown menu showing a list of groups: FAST_ADMIN, FAST_ANONYMOUS, FAST_AUTHENTICATED, FAST_HELP_EDITOR, Staff, -FAST Tech Support, -PLS Fast Admin, -Financial Aid Staff, -Loan Inquiry, -SRFS Accounting, -SRFS Counselor, -SRFS SWAT, -SRFS Staff, -Grad Rep, -Loan Office, -Disbursement Staff, -Loan Director, -Loan Manager, -Loan Office Admin Staff, -Senior Loan Officer, -System Administrator, -Help Manager, -System Administrator, -Tech Support, -IT Manager, -Tech Support Staff, customer, and -Student. Below the search box, there are fields for 'System name', 'Friendly name', and 'Hierarchical Path'. There are also radio buttons for 'Active' and 'Show users', and buttons for 'Commit' and 'Refresh'. The 'Name Pennic' is visible at the bottom of the group list.

This screenshot shows sets which are hierarchical bundles of privilege resources. In this case "caHostFind.jsp" is a resource which is a screen in the system, and it is in the "ca" set, which is in the "officeJSP" set.

Help Documentation Log out

Authorization - Edit/Delete Set

Find a FAST set to edit or delete
Set Name:

Env:
User: Hyzer, Michael C
Group: FAST_ADMIN

- Admin Console Home
- Back to application
- Authorization
 - Auth - export / import
 - Data - add
 - Data - edit / delete
 - Groups - add
 - Groups - edit / delete
 - Resource assignment
 - Resource report
 - Sets - add
 - Sets - edit / delete**
 - Users - add
 - Users - edit / delete
 - Users - import
- Audit Viewer
- CMS
- Community Logic
- Config Manager
- Daemons
- Data Table Editor
- Help System
- JSP compilation report
- Log download
- Messages
- System Monitor
- Logout
- View Reports
- Configure Reports

Edit an existing set

Name:

Parent Name: SET: officeJSP

Available:

- SET: mainPages
 - PAGE: mainPageLoanOfficer.jsp
 - PAGE: mainPageStudent.jsp
- SET: menuList
 - MENU: faMenu
 - MENU: grMenu
 - MENU: loMenu
 - MENU: saMenu
 - MENU: tsMenu

Children:

- SET: officeJSP
 - SET: app
 - SET: ca
 - PAGE: caHostContactAddressPop.jsp
 - PAGE: caHostFind.jsp
 - PAGE: caHostInstList.jsp
 - PAGE: caHostInstQueryPop.jsp
 - PAGE: caInfo.jsp
 - PAGE: caInfoDtl.jsp
 - PAGE: caSpecifyHostInst.jsp
 - PAGE: caUpdHostInst.jsp

Active Deleted

Copyright © 2006, University of Pennsylvania. All rights reserved.

This screenshot shows the view/assignment screen where you can assign privileges to a group, or to a user in a group.

FAST Admin Console

Env: User: Hyzer, Michael C Group: PLS Fast Admin

- Admin Console Home
- Back to application
- Authorization
 - Auth - export / import
 - Data - add
 - Data - edit / delete
 - Groups - add
 - Groups - edit / delete
 - Resource assignment
 - Resource report
 - Sets - add
 - Sets - edit / delete
 - Users - add
 - Users - edit / delete
 - Users - import
- Audit Viewer
- CMS
- Community Logic
- Config Manager
- Daemons
- Data Table Editor
- Help System
- JSP compilation report
- Log download
- Messages
- System Monitor
- Logout
- View Reports
- Configure Reports

Authorization - Property Value Assignment

Find a FAST group (or group and user) to view or assign

View by: Group / User Property Value

Group name:

User pennid (optional):

User pennkey (optional):

Normal access Admin access Both

View or assign property value

View / assign property values for group: "Loan Inquiry"

- ALLOW FORBID INHERIT
- ▶ SET:FAST_ADMIN_SET
 - ▶ SET:FAST_EDIT_HELP_ONLY_SET
 - ▶ SET:FAST_EDIT_HELP_SET
- ▶ SET:FAST_JSP_SET
- ▶ SET:PLSAdmin
- ▶ SET:StartJSPSet
- ▶ SET:admin
- ▶ SET:adminAccess
- ▶ SET:common JSP
- ▶ SET:devel
- ▶ MENU:fastAdminNonProd
- ▶ MENU:fastAdminProd
- ▶ SET:mainPages
 - ▶ PAGE:mainPageLoanOfficer.jsp
 - ▶ PAGE:mainPageStudent.jsp
- ▶ SET:menuList
- ▶ MENU:faMenu
 - ▶ MENUBUTTON:faMenu fastNone Query Penn Loan System

In this case, some entries are red and green (if allowed or forbidden). Most resources inherit permissions from a parent set or a parent group. There is an explicit assignment on this screenshot for the Loan Inquiry group to have access to mainPageLoanOfficer.jsp.

Since permissions can have three states (allow, forbid, unassigned [inherit]), then the decision making process to see if someone is allowed to do something is as follows:

1. See if there is an explicit permission for the user/group combo for the resource.
2. If so, done, if not, walk up the hierarchy of parent Set's until you find answer.
3. If not found, then do the same thing for the group in general (not user/group), then the parent group, etc.
4. If no privileges assigned, default is forbid.

That decision making logic is inside the framework in one place (though copied for each app). It is complex logic, and not something that we would want apps to have to reinvent.

There is an API to ask the framework if the current user has access to a resource:

```
if (Authorization.isAllowed(currentUser, FastPropertyType.CUSTOM_DATA, "org123")) {  
    ... whatever ...  
}
```

There are two "lists" (grouped term) for each privilege assignment, for access to the privilege resource, or to be able to admin (assign others) to the privilege resource. Anyone who has access to the admin screen (which is not a lot of people) can view all permissions (but perhaps not assign).

This screenshot shows the view of which groups or users have access to a specific resource. The last image shows ability to attach an expire date to the privilege.

The screenshot displays the 'FAST Admin Console' interface. On the left is a dark blue sidebar with the University of Pennsylvania logo and a navigation menu. The main content area is titled 'Authorization - Property Value Assignment'. At the top, there are tabs for 'Help' and 'Documentation', and a 'Log out' button. Below the title is a search box: 'Find a FAST group (or group and user) to view or assign'. It includes a 'View by:' dropdown set to 'Property Value', a 'Property value:' dropdown set to '-SET: trackingCert01', and radio buttons for 'Normal access', 'Admin access', and 'Both'. A 'View or assign property value' button is below. The main section is titled 'View / assign property values for property value: "SET: trackingCert01"'. It features a tree view of groups and users with icons for 'ALLOW', 'FORBID', and 'INHERIT'. A 'Submit Changes' button is at the bottom. Below the tree is a section for 'Set expire date for property value assignment for property value: "SET: trackingCert01"', with columns for 'Group - User' and 'Expire date'. A footer contains the copyright notice: 'Copyright © 2006, University of Pennsylvania. All rights reserved.'

On the "resource report" screen, the framework will automatically parse all the JSPs to sync up the JSP/menu/buttons available for assignment, so these do not need to be manually entered.

This screen shows how expiration dates can be applied to assignments

The screenshot shows a web application interface. On the left is a dark blue sidebar with navigation links: Logout, View Reports, and Configure Reports. The main content area has a list of assignments with status icons (triangles and circles) and names like SET:mainPages, SET:menuList, SET:officeJSP, MENU:stMenu, SET:studentJSP, MENU:tpMenu, SET:trackingCert, SET:trackingUpd, and SET:trash. Below this list is a 'Submit Changes' button. The bottom section is titled 'Set expire date for property value assignment for group: "Senior Loan Officer"'. It contains a table with columns for 'Property Value Name' and 'Expire date'. The table lists several property values and their corresponding expire dates, with a 'Set Expire Date' button for each row.

Property Value Name	Expire date
SET: FAST_EDIT_HELP_ONLY_SET	12/01/2009
PAGE: mainPageLoanOfficer.jsp	
PAGE: mainPageStudent.jsp	
MENU: loMenu	
MENUBUTTON: FASTBloMenu fastNone Manage disbursement	
MENUBUTTON: FASTBloMenu menuCancel conProcButton Cancel processed loan	

It does include hooks, auditing, web services, import/export xml, etc... generally we have used this authorization system for access to webapp resources (buttons, menus, jsps, etc), though we also use it for custom permissions (e.g. which data a user can see).