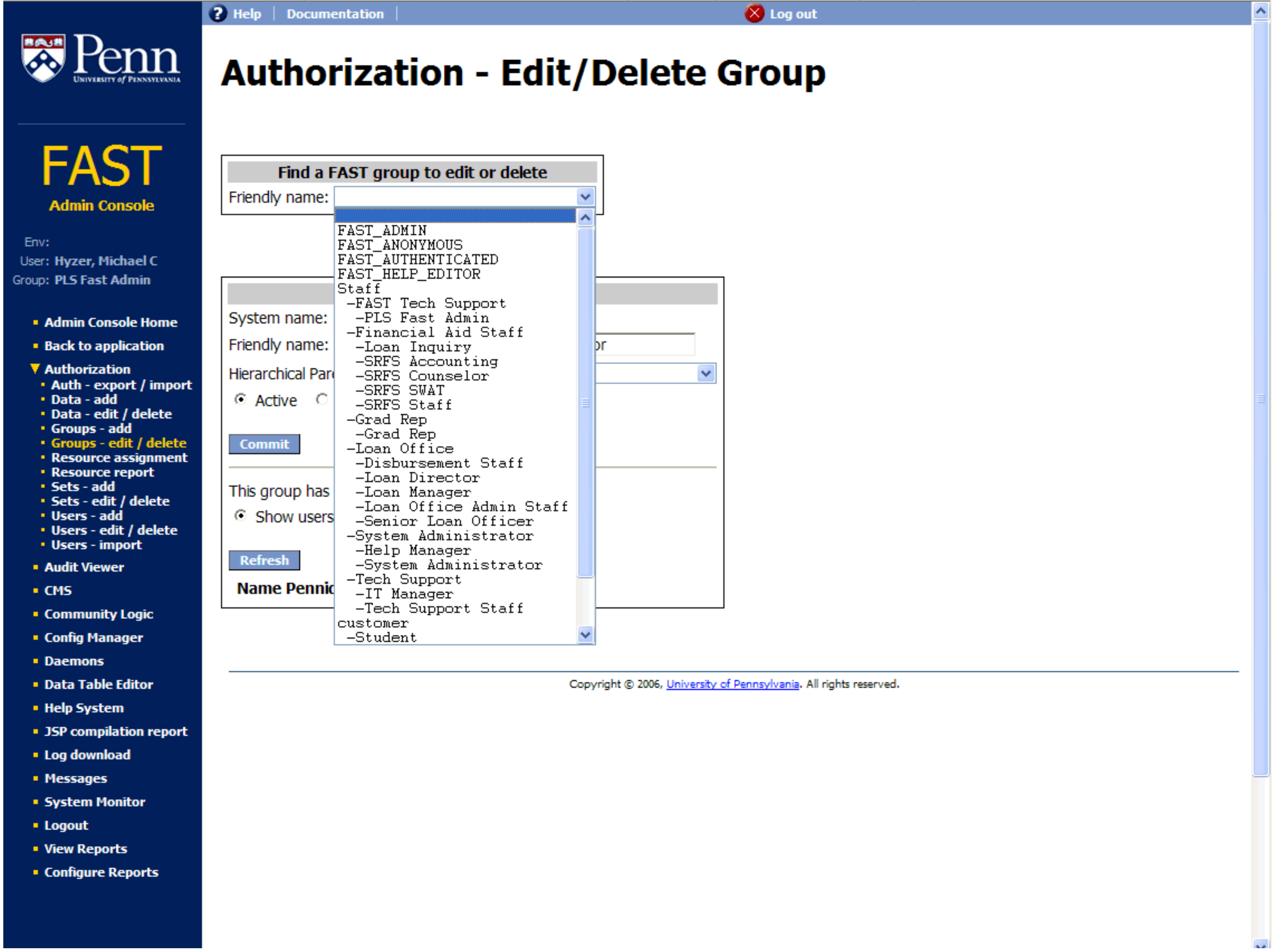
Penn authorization system

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.



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

Authorization - Edit/Delete Group

Find a FAST group to edit or delete

Friendly name:

```
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
  -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
 -Student
```

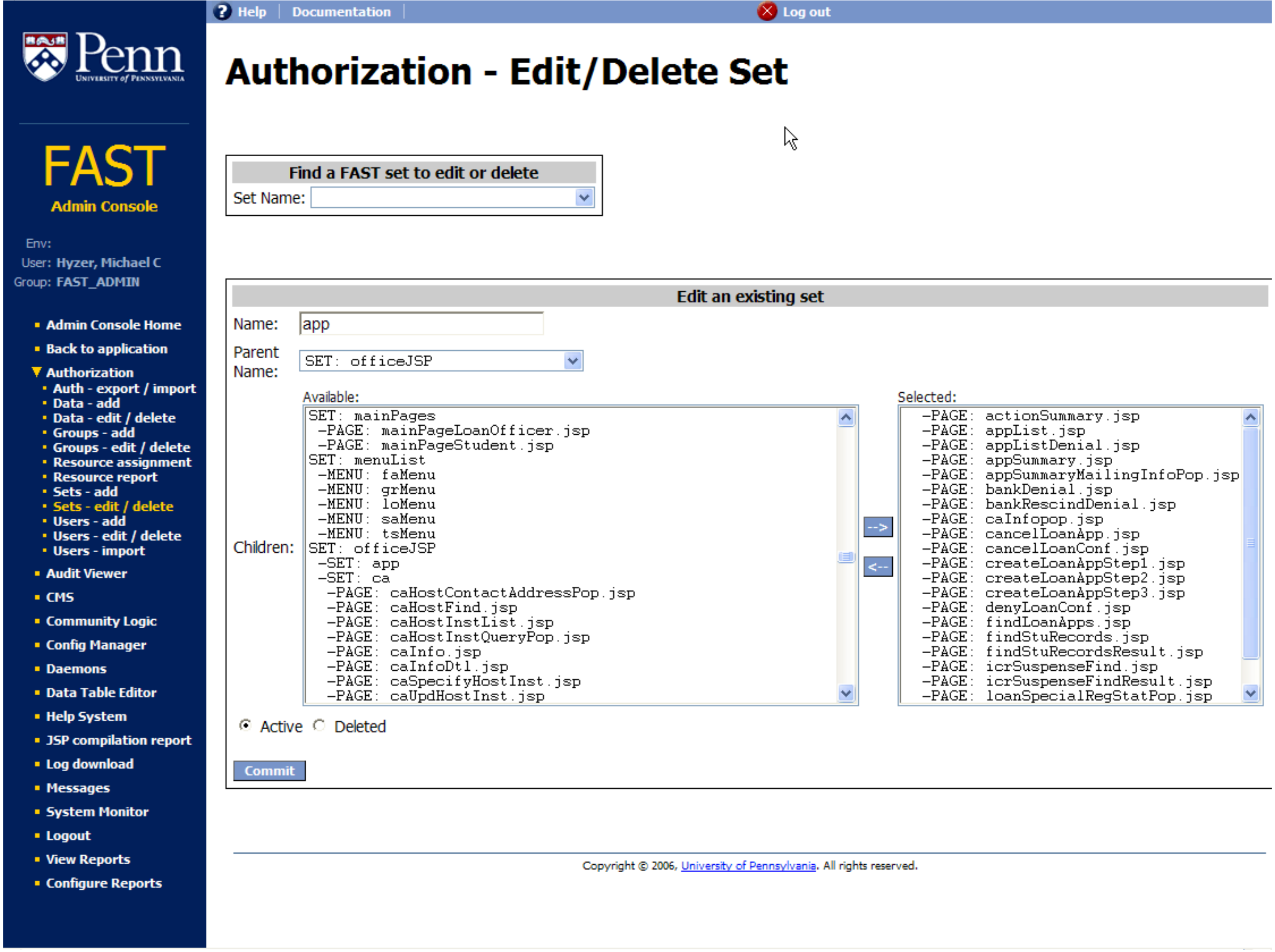
System name:
Friendly name: or
Hierarchical Par
● Active ○

This group has
● Show users

Name Pennic

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

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.



Env:
User: Hyzer, Michael C
Group: FAST_ADMIN

Authorization - Edit/Delete Set

Find a FAST set to edit or delete

Set Name:

Edit an existing set

Name: app

Parent Name: SET: officeJSP

Children:

Available:

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

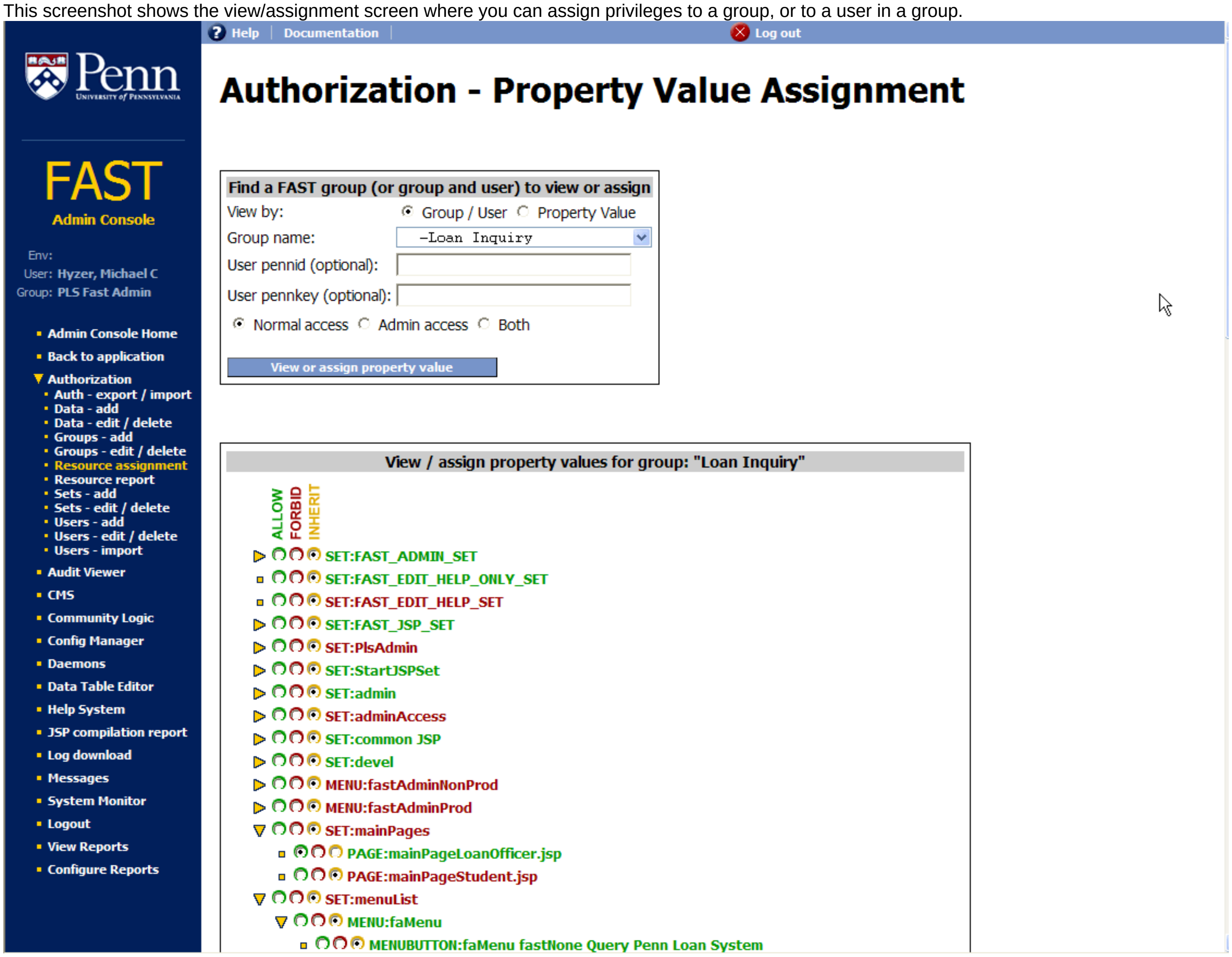
Selected:

```
  -PAGE: actionSummary.jsp
  -PAGE: appList.jsp
  -PAGE: appListDenial.jsp
  -PAGE: appSummary.jsp
  -PAGE: appSummaryMailingInfoPop.jsp
  -PAGE: bankDenial.jsp
  -PAGE: bankRescindDenial.jsp
  -PAGE: caInfopop.jsp
  -PAGE: cancelLoanApp.jsp
  -PAGE: cancelLoanConf.jsp
  -PAGE: createLoanAppStep1.jsp
  -PAGE: createLoanAppStep2.jsp
  -PAGE: createLoanAppStep3.jsp
  -PAGE: denyLoanConf.jsp
  -PAGE: findLoanApps.jsp
  -PAGE: findStuRecords.jsp
  -PAGE: findStuRecordsResult.jsp
  -PAGE: icrSuspenseFind.jsp
  -PAGE: icrSuspenseFindResult.jsp
  -PAGE: loanSpecialRegStatPop.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.



Help | Documentation | Log out

Penn University of Pennsylvania

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: -Loan Inquiry

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:

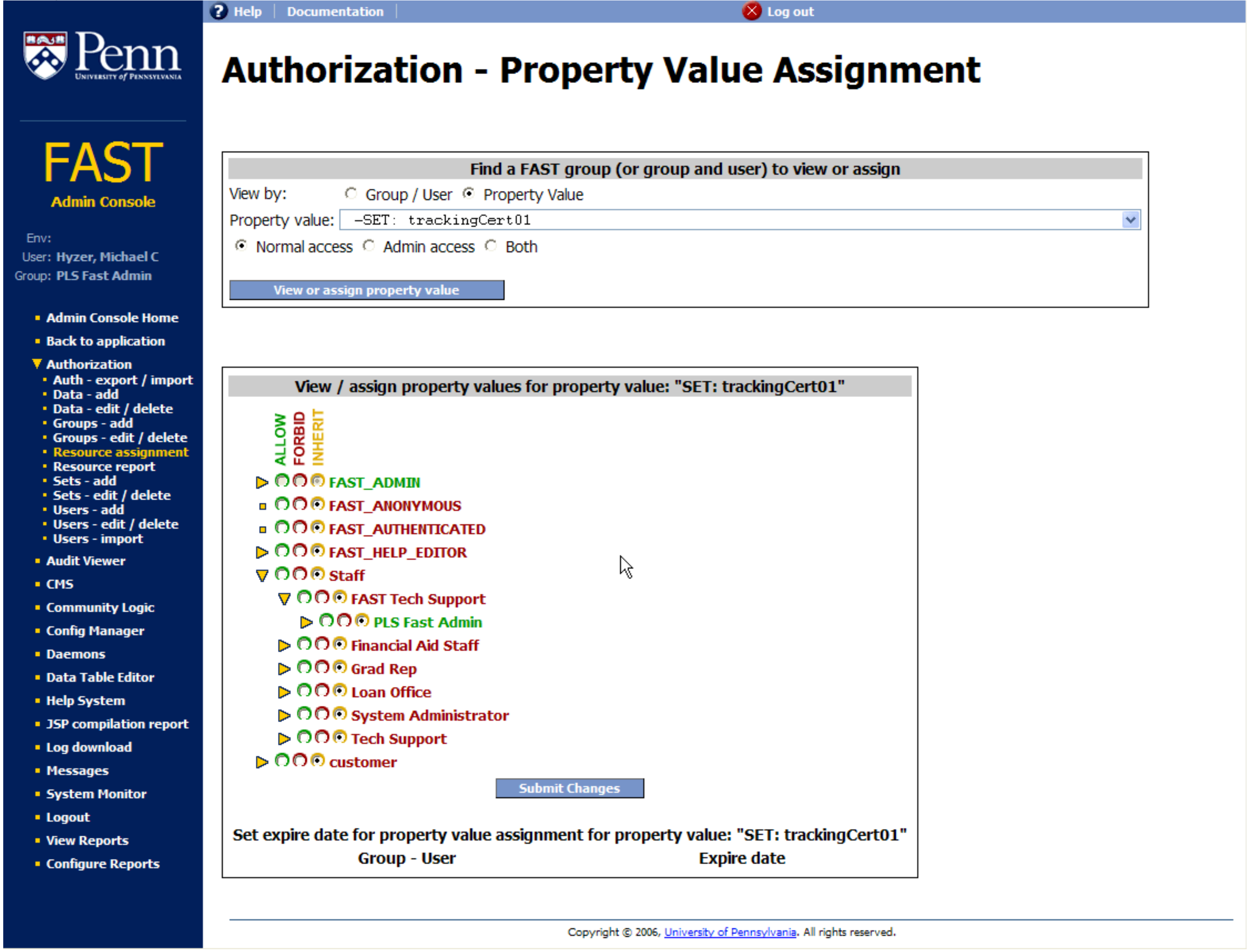
1. See if there is an explicit permission for the user/group combo for the resource.
2. If so, done, if not, walk up the hierarchy of parent Set's until you find answer.
3. If not found, then do the same thing for the group in general (not user/group), then the parent group, etc.
4. If no privileges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



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

Authorization - Property Value Assignment

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

View by: ○ Group / User ● Property Value

Property value: -SET: trackingCert01

● Normal access ○ Admin access ○ Both

View / assign property values for property value: "SET: trackingCert01"

ALLOW FORBID INHERIT

```
▷ ○ ○ ● FAST_ADMIN
□ ○ ○ ● FAST_ANONYMOUS
□ ○ ○ ● FAST_AUTHENTICATED
▷ ○ ○ ● FAST_HELP_EDITOR
▽ ○ ○ ● Staff
    ▽ ○ ○ ● FAST Tech Support
        ▷ ○ ○ ● PLS Fast Admin
    ▷ ○ ○ ● Financial Aid Staff
    ▷ ○ ○ ● Grad Rep
    ▷ ○ ○ ● Loan Office
    ▷ ○ ○ ● System Administrator
    ▷ ○ ○ ● Tech Support
▷ ○ ○ ● customer
```

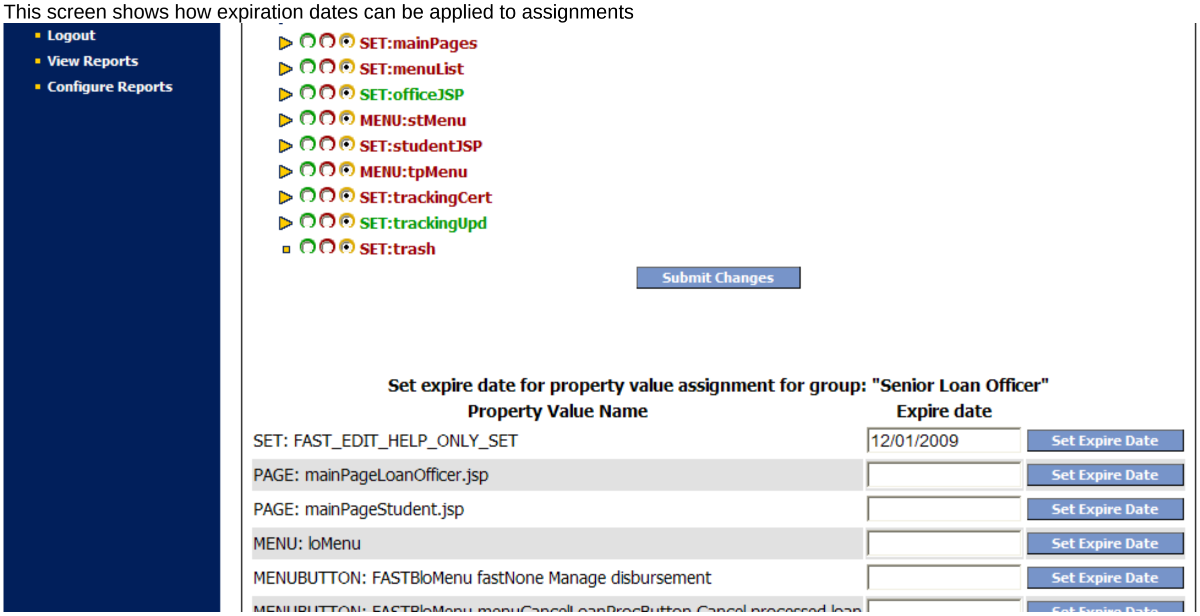
Set expire date for property value assignment for property value: "SET: trackingCert01"

Group - User	Expire date

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



- SET:mainPages
- SET:menuList
- SET:officeJSP
- MENU:stMenu
- SET:studentJSP
- MENU:tpMenu
- SET:trackingCert
- SET:trackingUpd
- SET:trash

Set expire date for property value assignment for group: "Senior Loan Officer"

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

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).