FU Berlin - Using Hooks

Defining own hooks

Step 1: Copying & pasting Grouper hooks to own java package

The original Grouper hooks are located in [Grouper API installation directory]/src/grouper/edu/internet2/middleware/grouper/hooks. The root is the edu level, thus having edu.internet2.middleware.grouper.hooks as the package name.

We have introduced an additional class hierarchy [Grouper API installation directory]/src/grouper/newLevel1/newLevel2/hooks.

Now copy & paste all original Grouper hooks, rename the class names to "MyXXXHook.java" (or anything) and modify the package name of each class to newLevel1.newLevel2.hooks..

Step 2: Make sure the new hooks are found

Open the file [Grouper API installation directory]/conf/grouper.base.properties and copy all of the

```
hooks.xxx.class=edu.yourSchool.it.YourSchoolxxxHooks,edu.yourSchool.it.YourSchoolxxxHooks2
```

lines into grouper.properties configuration file.

Now register the new classes by modifying the new entries:

```
hooks.attribute.class=newLevel1.newLevel2.hooks.MyAttributeHook
```

```
hooks.attributeAssign.class=newLevel1.newLevel2.hooks.MyAttributeAssignHook
```

(and so on for all hooks)

Step 3: Customize your hooks

By default the hooks are empty but you can use them easily for triggered actions. All hooks are executed automatically if the corresponding activity happens. I.e. the MyGroupHook.groupPostUpdate() method will be executed after each group update. Now fill them with life.
Here are two examples of how we use them.

Sending mails if important member groups are deleted

We want to be notified if source groups that are direct members of target groups are deleted. For this the MyMembershipHook class is the right choice. Here edit the membershipPostDelete() method.

```
public void membershipPostDelete(HooksContext hooksContext, HooksMembershipBean postDeleteBean) {

HooksMembershipChangeBean membershipChangeBean = new HooksMembershipChangeBean (postDeleteBean.getMembership());
// get membership informations

Properties properties = System.getProperties();
properties.setProperty("mail.smtp.host",
"yourHost");
// smtp host

Session session = Session.getDefaultInstance(properties);

if (postDeleteBean.getMembership().getOwnerGroupId() != null ) {

        if ( membershipChangeBean.getGroup().getName().startsWith ("targets:")  &&                        //
owner group is in target folder
             membershipChangeBean.getMember().getName().startsWith ("sources:") &&                        //
member is in source folder
             membershipChangeBean.getMember().getSubjectSourceIdDb().equals ("g:gsa") ) {                 //
member is a group

             try {
                  MimeMessage message = new MimeMessage(session);
                  message.setFrom(new InternetAddress("sender@yourEdu.
xx"));                                    // sender
                  message.addRecipient(Message.RecipientType.TO, new InternetAddress("recipient@yourEdu.
xx"));    // recipient
                  message.setSubject( "Grouper Group Removal
Notification");                                                                          // subject

                  message.setText ("-------------------------------------------\n"
+                           // message body
                                 " Grouper Group Removal Notification \n" +
                                 "-------------------------------------------\n\n" +
                                  membershipChangeBean.getMember().getName()
+                                                          // name of member group
                                 "\n\nwas removed from\n\n" +
                                  membershipChangeBean.getGroup().getName()
+                                                             // name of owner group
                                 "\n\n");

                Transport.send(message);

          } catch (MessagingException mex) {
                mex.printStackTrace();
          }
       }
    }
  }
```

Setting privileges if groups are created within a certain stem

If groups are created within a certain stem some privileges are to be set automatically. This is done with the groupPostCommitInsert() method in MyGroupHook.

```
public void groupPostCommitInsert(HooksContext hooksContext, HooksGroupBean postCommitInsertBean) {

    Group beanGroup = postCommitInsertBean.getGroup();
// get group informations

    /* Read-Privilege for staff for all target groups */

    if ( beanGroup.getName().startsWith("targets:"))
{
// all target groups
        String employersGroupId = GroupFinder.findByName (GrouperSession.startRootSession(), "stem:
group_of_employees", false).getId();                        // id of staff group
        beanGroup.grantPriv ( SubjectFinder.findById ( employersGroupId, false)  , AccessPrivilege.READ
);                                                                                  // granting read privilege
     }

     /* Admin Privileges for xGroup for all myNet groups (i.e. groups that contain the string "myNet" in their
full name */

     if ( beanGroup.getName().contains( "myNet" ))
{
// all groups that contain "myNet"
         String xGroupId = GroupFinder.findByName (GrouperSession.startRootSession(), "stem:xGroup", false).
getId();                                                    // id of xGroup
         beanGroup.grantPriv ( SubjectFinder.findById ( xGroupId, true), AccessPrivilege.ADMIN
);                                                                                                  // granting
admin privilege
     }
 }
```

Writing log entries into syslog if members or groups are deleted/added within a certain stem

Using MyMembershipHook. We want to log all membership changes of groups within the stem "xxx:myNet" into syslog.

```
/* Adding a member */
public void membershipPostAddMember(HooksContext hooksContext, HooksMembershipChangeBean postAddMemberBean)   {

    if ( postAddMemberBean.getMembership().getOwnerGroupId() != null )
{                                                                    // could also be a stem as owner,
but we're just interested in groups and their members

       if (postAddMemberBean.getGroup().getName().contains ("myNet")
&&                                                                                  // the
stem we want to monitor; the "name" of the group contains the entire path
           postAddMemberBean.getMembership().getFieldId().equals (FieldFinder.find ("members", true).getUuid()))
{         // we don't want to get privs, just real members

                        logToSyslog (hooksContext, postAddMemberBean,
"Added");
// logging
      }
   }
}

/* Removing a member */
public void membershipPostRemoveMember(HooksContext hooksContext, HooksMembershipChangeBean
postDeleteMemberBean) {

    if ( postDeleteMemberBean.getMembership().getOwnerGroupId() != null )
{                                                                    // could also be a stem as owner,
but we're just interested in groups and their members

       if (postDeleteMemberBean.getGroup().getName().contains ("myNet")
&&                                                                                  // the stem we
want to monitor; the "name" of the group contains the entire path
           postDeleteMemberBean.getMembership().getFieldId().equals (FieldFinder.find ("members", true).
getUuid())) {      // we don't want to get privs, just real members

         logToSyslog (hooksContext, postDeleteMemberBean,
"Removed");
// logging
      }
   }
}
```

The method logToSyslog() is defined as additional function:

```
private static final String SYSLOG_SEVERITY =
"info";
// e.g. Error, Debug, Warning ...
private static final String SYSLOG_FACILITY =
"local3";
// e.g. local0, local1, local2 ...

private void logToSyslog (HooksContext context, HooksMembershipBean bean, String action) {

    String msg = "(Grouper) "
+
// composing message text
                 "Group \"" +  bean.getMembership().getOwnerGroup().getExtensionDb() + "\": " +
                  action + " " + bean.getMembership().getMember().getSortString1() +
                 " [" +  bean.getMembership().getMember().getName() + "]" +
                 " / Initiated by: " + context.getSubjectLoggedIn().getId();
      try {

          new ProcessBuilder( "logger", "-p", Integer.toString(Integer.parseInt(SYSLOG_FACILITY +
SYSLOG_SEVERITY, 2)), "\"", msg , "\"" ).start();  // calling command line function "logger" with arguments


      } catch ( IOException e ) {
         LOG.error ("Syslog could not be written in " + MyMembershipHook.class.getName());
      }
}
```

Here's an example of the resulting log entry:

```
Mar 21 12:59:28 9e:myServer (Grouper) Group "myGroup": Added Doe, John [jdoe] / Initiated by: myName
```

Supplying unixGroups with gidNumbers

This is currently done with attributes from Grouper 1.x that were turned into legacyAttributes with the latest Upgrade. We are currently working on transferring our solution into the new Attribute Framework.