

# Penn email list provisioner

<a href="#">Wiki Home</a>	<a href="#">Grouper Release Announcements</a>	<a href="#">Grouper Guides</a>	<a href="#">Grouper Deployment Guide</a>	<a href="#">Community Contributions</a>	<a href="#">Internal Developer Resources</a>
---------------------------	---	--------------------------------	--	---	--

## Leveraging Grouper's New Features to Improve Handling of Email Lists at University of Pennsylvania

At Penn we use LSoft Listserv for email lists. We recently had a project to make it easier to manage automated lists and to improve deprovisioning of list owners and members. Using Grouper to solve these problems has allowed us to address the shortcomings of the previous service while using new features of the Grouper platform.

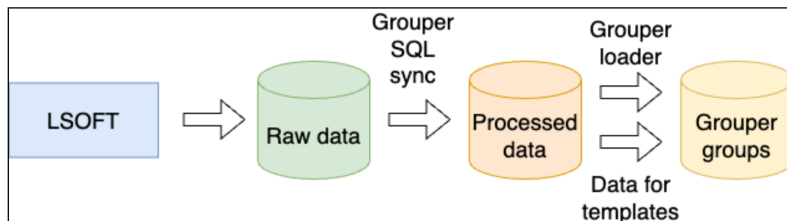
There are four email list roles that we are tracking in Grouper: owners, quiet owners, senders, and members. To make this easier to explain, we will just discuss owners and members, but the quiet owners and senders have similar characteristics as owners. Not to mention the typical goal of retiring bespoke unmaintainable perl scripts!

### Integration

We have two feeds for LSoft: one that sends current lists, owners, and members to Grouper, and one that sends members and owners of Grouper managed lists back to LSoft. Grouper integrates via SQL for both of these, and the LSoft team implemented scripts to integrate with LSoft.

### User Feed from LSoft

Data from LSoft to Grouper is needed for several reasons: we need to know if a list is new or not when configuring in Grouper, we need to know that the Grouper managed list corresponds to what was in LSoft for existing lists, and we want to report on LSoft lists that are not managed in Grouper. This diagram shows the data flow:



This feed is sent to two tables that Grouper can read: the names of the email lists, and the members and roles. One challenge is that users for the lists are email addresses, which are difficult or impossible to match to institutional users. These are matched by: seeing if the email address is associated with the user in the directory, seeing if the email address matches the EPPN (e.g. [jsmith@institution.edu](mailto:jsmith@institution.edu)), or seeing if it is our domain and the NetId matches the email prefix. This match is done in a view against all emails from LSoft, but it is too slothy for frequent use, so a Grouper SQL Sync is used to copy the results to an indexed table.

Grouper can sync from SQL (e.g. a view or table in any database), LDAP, web service, etc to a SQL table. The synchronization can be real time or batch. These ETL (extract, transform, load) tasks are very useful for integrations since they can move data to where you need it, or simplify complex operations into indexed results. Some opportunities with the new system are that users can see the overall view of their lists no matter what email address they used, and as they change their address in the directory it will be reflected in all email lists that Grouper manages!

Once the email addresses are matched to identities (if they can be), Grouper can use those in Grouper groups or template processes (discussed later). If email list owners do not want to manage a list in Grouper for whatever reason, they can still load the list roles into Grouper groups with the Grouper Loader. Then they can get notifications, use attestation, see point-in-time memberships, generate reports, and other features available to Grouper groups. Eventually they might change their mind!

### Group Structures

The group structures for the list ensure that all the roles are accounted for: automated memberships (usually [reference groups](#)), manual memberships, and external non-person email addresses (e.g. a list is a member of another list), but lists are not Subjects in Grouper!). More could be written just on this topic, but we will just briefly describe it here.

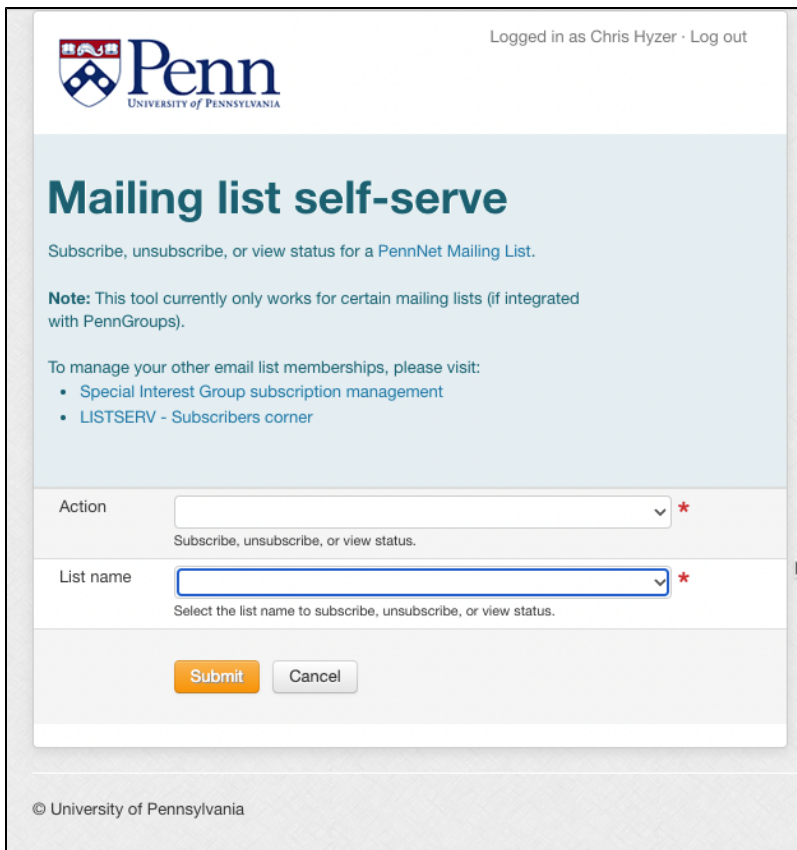
it-announce_list
it-announce_list_automatic
it-announce_list_editors
it-announce_list_exclude_people
it-announce_list_external
it-announce_list_include
it-announce_list_include_people
it-announce_owners
it-announce_owners_external
it-announce_owners_people
it-announce_owners_people

Do not be afraid of this large number of groups; the templates will be a nice facade for them. There are four groups for the four roles. There is an editors group for people who do not own the list, but who can manage memberships (e.g. help desk). The includes and excludes for people are for opt ins / opt outs, or a help desk person to intervene. The automatic group has reference groups inside (e.g. org groups or people who are IT staff). The external group has email addresses not represented as subjects. This is done with [Grouper Local Entities](#). These are objects that have security about who can use them. Then they can be used by provisioning, the UI, [GSH \(Grouper Shell Script\) templates](#), etc.

For the manual groups, Grouper Membership Requirements are used to ensure that when people no longer meet the requirement they are automatically removed. A recent Grouper enhancement is the ability to edit membership requirements from the “Edit group” screen. Easier deprovisioning leads to more frequent deprovisioning.

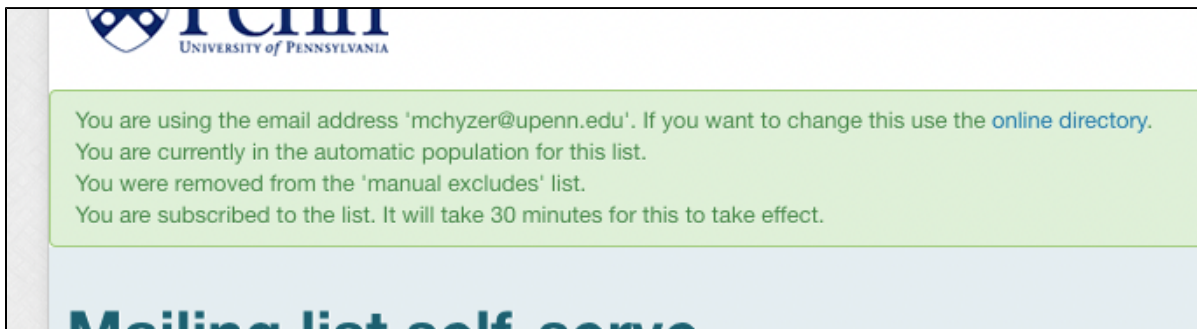
## User Self Service

If a list allows it, and the user is in the eligible population, users can opt in to a list. Or maybe they can just opt out. This process is a little complex and requires a friendly UI. An example of the complication is when a user wants to opt in to a list. If they are in the automatic population, and in the opt outs list (due to previously opting out, or for some other reason), then the correct action is to remove them from the opt outs list. But this is not intuitive for end users and would be time consuming for support staff to do on their behalf. So a [Grouper GSH template](#) is used to make a UI to do the correct logic. A recent enhancement to GSH templates is the "simplified UI" which removes all unneeded Grouper UI stuff and leaves only the part that the user needs. The drop down of lists to subscribe to only has lists that the user is allowed to opt in to.



The screenshot shows a web interface for the University of Pennsylvania. At the top, it says "Logged in as Chris Hyzer · Log out". Below the Penn logo, the title "Mailing list self-serve" is displayed. The main heading is "Subscribe, unsubscribe, or view status for a PennNet Mailing List." A note states: "Note: This tool currently only works for certain mailing lists (if integrated with PennGroups)." Below this, it says "To manage your other email list memberships, please visit:" followed by two links: "Special Interest Group subscription management" and "LISTSERV - Subscribers corner". The form has two dropdown menus: "Action" with a red star icon and "List name" with a red star icon. Below the "List name" dropdown is the text "Select the list name to subscribe, unsubscribe, or view status." At the bottom of the form are "Submit" and "Cancel" buttons. The footer says "© University of Pennsylvania".

The result of the screen is adding or removing the user to or from the necessary groups. This is the result of opting in to a list when previously being in the opt outs.



The screenshot shows a confirmation message box with a green background. It contains the following text: "You are using the email address 'mchyzer@upenn.edu'. If you want to change this use the [online directory](#). You are currently in the automatic population for this list. You were removed from the 'manual excludes' list. You are subscribed to the list. It will take 30 minutes for this to take effect." Below the message box, the title "Mailing list self-serve" is partially visible.

There is a test and prod LSoft system and a test and prod folder in Grouper for these mailing lists. A recent enhancement to GSH templates allows the same template code to be used in two templates, and key off of the template ID to know if it is test or prod and adjust the queries or folders accordingly. This makes it easy to add enhancements and migrate code from the test to prod template.

## Help Desk List Management

Help desk workers or other people who manage lists need to be able to achieve a similar function but on behalf of other users. There is a GSH template for them to accomplish this, but only for the lists that they own or manage.

### Mailing list admin

Administer user lists for a PennNet mailing list

Action	<input type="text" value="Subscribe"/>	*
Subscribe, unsubscribe, view status for a list, view status for all lists.		
List name	<input type="text" value="safedns-lsps"/>	*
Select the list name to subscribe or unsubscribe		
Type	<input type="text" value="Person"/>	
Type of addition, external email address or person		
Pennkey or pennid	<input type="text"/>	*
Enter the PennKey or PennID of the user to manage. Can be multiple, comma-separated values		

Since the inputs, logic, validation, and messaging are similar for the self-service vs admin interfaces, some of the same methods in the template are re-used in both templates.

## Requesting a New List

When a new list or migration of an existing list is requested, a GSH template allows the settings to be captured and sent to the email list team for review. The activation of this list can be delayed so the members can be verified before cutover. Do not activate the list until you are absolutely ready!



When the create request is submitted, it will store the configuration settings in a database table, assign a request ID to it, and send an email to the email list team to review. When the team approves it, the Grouper folders, groups, memberships (based on feed from LSoft), membership requirements, and provisioning settings are automatically created.

This shows an approval for a list named “abc” submitted by me:

**PennNet mailing list team**  
Actions that mailing list team can do

Action	createListRequestAction	★
Action		
Request	abc - 5af6 - 23-11-27 15:16 - Chris Hyzer	★
Which request		
List create request action	create	★
Should this request be implemented or not		

SubmitCancel

## Provisioning the Lists to LSoft

The new provisioning framework facilitates real time and nightly full sync provisioning of the email roles to LSoft. There are many supported provisioning targets, but in this case we are provisioning the SQL tables. Provisioning “metadata” is information on the group (or entity) which informs the target of characteristics about the group. Here is an example for an email list member role:

Group provisioning settings Provisioning actions ▼

Target name	listservIntegration	★
Target where you want to provision		
Type	Yes, has direct provisioning configuration	
If this group/folder has provisioning configuration directly assigned		
Provision	Yes, provision the object	
If this object should be provisioned or not		
Allow optout	<input type="radio"/> Default value (False) <input checked="" type="radio"/> True <input type="radio"/> False	
If users should be able to opt out if there is an opt out group		
Listserv prefix	it-announce	★
Enter the first part of the listserv name. e.g. is the email address is some_email_list@lists.upenn.edu, then enter: some_email_list		
Enabled	<input type="radio"/> Default value (False) <input checked="" type="radio"/> True <input type="radio"/> False	
If true this is sent to the integration. If false you can setup in penngroups but it wont affect the LSOFT list yet		
List description	IT announce list	
The description of the list in lsoft listserv. Note, the only description used will be from the members list, so do not add this to the ov		
List to manage	members	★
Which list to manage of this list in listserv		
Optin allow group name	penn:community:activeNonAlumniWithPennname	
Group of users who are allowed to opt in		

SaveCancel

In this case the metadata is filled out automatically by the GSH template after the approval so users do not need to be bothered with it, but in other provisioning use cases, users might be setting things like this on provisionable objects.

## Summary

Rolling out this Grouper integration with LSoft is in its early phases and so far the experience has been positive. When we have integrated more lists and kicked the tires a little more we will post specifics on how things are configured and the template code needed to make this work. Stay tuned on the uncommon-grouper slack channel!