

# **Academus<sup>®</sup> Portal Template Users Supplement**

**Version 2.1**

***CONFIDENTIAL***

# Notices

This Unicon® Academus® Portal Template Users Supplement document contains important system and operational information related to Academus.

Use of Academus in conjunction with other products and services, other than those expressly authorized by Unicon, Inc. is the user's sole responsibility.

Due to continuing product development, the specifications and capabilities described in this document are subject to change without notice.

## **DISCLAIMER**

Unicon, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing or use of this document.

## **CONFIDENTIAL**

All information contained herein is confidential and proprietary to Unicon, Inc. and constitutes trade secrets of Unicon, Inc. This document shall not be disclosed, duplicated or used, in whole or in part, for any purpose other than its authorized use with the Academus product. The information contained herein shall not be communicated to a third party without prior written consent from Unicon. Any unauthorized use is strictly prohibited.

## **TRADEMARKS AND SERVICE MARKS**

Unicon® and Academus® are a registered trademarks of Unicon, Inc.

All trademarks, service marks and trade names referenced herein are the property of their respective owners. Any use, copying or reproduction of the Unicon, Inc. service marks without prior written permission of Unicon is strictly prohibited.

## **COPYRIGHT**

© 20067Unicon, Inc., Chandler, Arizona. All Rights Reserved.

## **LICENSE AGREEMENTS**

The Academus product includes software products that are licensed by manufacturers other than Unicon. Your use of the Academus service indicates your acceptance and agreement to comply with the terms of such license agreements provided with Academus.

# Table of Contents

<b>Section 1. Introduction</b>	<b>5</b>
Purpose	5
Target Audience	5
Assumptions	5
Definitions	6
<b>Section 2. Understanding the Key Portal Concepts</b>	<b>7</b>
Templates	7
Benefits of Using Templates	7
What are Template Users?	7
What Are the Default Template User Settings?	8
Why is the Template User “demo” Important?	8
Why is the Template User “guest” Important?	8
Why Have Different Template Users?	8
Fragments	9
Understanding Fragments	9
Benefits of Using ALM fragments	10
What Are ALM Fragments?	10
PAGS	11
Benefits of Using PAGS	11
What is PAGS?	11
Enabling PAGS	11
<b>Section 3. Procedures</b>	<b>13</b>
Creating and Configuring User Templates	13
Step 1: Create the Template User	13
Step 2: Configure the Template User’s Layout	13
Step 3: Assign the Template User to the Appropriate Group(s)	13
Step 4: Configure Academus to Utilize Template Users	13
Step 5: Verify Successful Completion of the Template User Setup	14
Managing Template Users	14
Retrieving Template User Values	14
Visible Changes	14
Breaking Affiliation with Template Users	15
Resetting the User Layouts	15
Changing the Default Template Users	15
“demo” Template User	15
“guest” Template User	16
<b>Section 4. Summary and Best Practices</b>	<b>17</b>
Managing Academus Permissions and User Layouts	17
PAGS Best Practices	18
Resetting All User Layouts	19
Resetting User Layout	21

## Notes

---

## Section 1. Introduction

This section describes the purpose, target audience, user assumptions, and definitions that are applicable to this document.

---

### Purpose

It is necessary to understand the key portal concepts, including Template Users, Person Attribute Group Service (PAGS), and fragments when using groups and layouts within Academus. This document assists in defining and understanding how to use these key portal concepts within Academus.

---

### Target Audience

This document is intended for use by portal and User Attribute Source administrators who use Academus 2.0. This document is used as a supplement to the current Academus documentation.

---

### Assumptions

In order for this document to be applicable, the following assumptions should be made:

- Your institution utilizes user attribute sources.
- The configuration discussed in this document has Academus looking to the user attribute source prior to checking the portal database for user information, such as the template user attribute.
- Six template users have been rolled out with your current configuration of Academus: faculty, staff, admin, students, demo, and guest.
- The layout, colors, etc. for your institution may differ from the layouts provided as examples in this document.

## Definitions

User Attribute Sources	The User Attribute Sources can include LDAP (Lightweight Directory Access Protocol) directories and relational databases.
Template Users	These type of users typically represent roles within the portal such as “student” or “faculty”, representing default layout content and local group memberships for user in this role. Each real user is mapped to one template user.  It also provides an example of how other user accounts could be initially configured.
Template User Types	The following template users are available in the baseline Academus deployment: admin, staff, student, faculty, demo, and guest.
Guest Template	The guest template user defines the unauthenticated layout; the page seen prior to logging into the portal.
Demo Template	The demo template user defines the layout for portal users without a known template user attribute value.
Layout	The selection and arrangement of content a user experiences in Academus portal is referred to as a layout. A given user's layout consists of applicable ALM fragments alongside personal layout content. The initial personal layout content of any user is the layout of the associated template user. The selection of ALM fragments included in any given user's layout is determined by his or her group memberships and preferences.
Pushed Fragments	Pushed fragments are fixed tabs that contain channels/portlets controlled by the administrator and cannot be edited by the end user.
Pulled Fragments	Pulled fragments are optional user-selected tabs that contain channels/portlets controlled by the administrator and cannot be edited by the end user.
PAGS	The Person Attribute Group Service defines portal group membership for users based upon the values in the User Attribute Sources.
Attribute	A property or value associated with a user for database purposes.
Authentication	Authentication is generally the verification of user credentials to determine user identity.
Authorization	Authorization occurs when LDAP or the portal database confirms that the user logging in is actually a “known” user and returns certain information to the portal as a result
Groups Manager	Administrative channel within Academus that enables particular users to determine and manage groups and permissions.
Group	A predefined collection of permissions that a user may belong to.

## Section 2. Understanding the Key Portal Concepts

This section provides the benefits and a brief explanation of Templates, Fragments and PAGS.

### Templates

#### Benefits of Using Templates

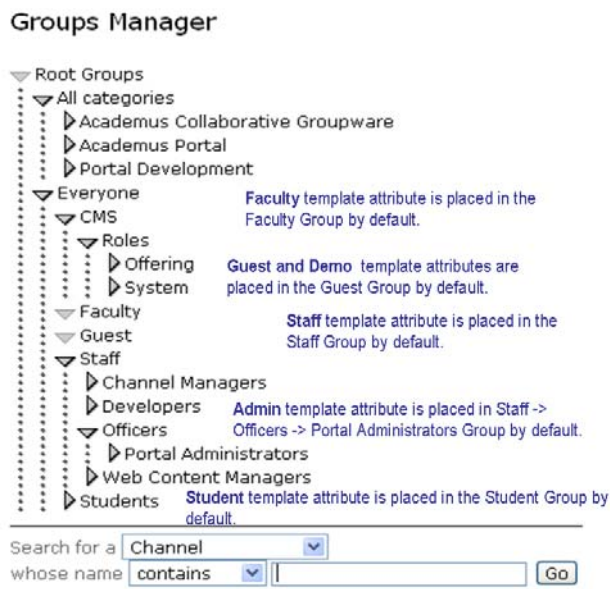
- Provides default local group memberships
- Provides single cohesive template layout for each user
- Provides end-user customizability

*Note: The portal database does not accommodate template users out of the box.*

#### What are Template Users?

Several types of user accounts exist within a completed Academus installation, including end users, students, staff, faculty, and other stakeholders who make use of the portal. A few accounts will not correspond to a specific user; these are administrative account types known as Template Users. The Template User account serves as the example for how other user accounts can be initially configured. Template users serve two purposes: They define portal layout and they provide default group affiliation.

Template users pre-populate the group affiliation of users associated with any particular template user. The group memberships of a template user define the initial group memberships of new portal users. The Portal Administrator can log in as one of the template users and then define a layout and default group affiliation for portal users.



*Note: Templates do not need to be active because they are part of the baseline deployment.*

## What Are the Default Template User Settings?

During the Academus deployment, it is necessary, as well as beneficial, to define the default groups/tabs/channels for a maximum of six template users. If you accept the default template user setups, the default template user setup, the following “standard” template users will be applied to Academus:

- admin
- student
- faculty
- staff
- demo
- guest

*Note: The template users are available to use/update/manage as soon as your deployment is complete. Unicon will provide passwords for each of these users.*

The first four template users listed (admin, student, faculty, staff) are the typical campus user types that are usually required by an educational institution. By default, the uPortal framework requires the existence of the last two template users (demo and guest).

## Why is the Template User “demo” Important?

A user who does not have a template attribute value is placed in the Demo group by default. For example, if the template attribute is not populated or contains a value that is not the name of a valid template user, the user will receive the layout and group assignment of the template user “demo”.

Demo is the default user layout designated in the uPortal framework; however, this setting can be configured. For details, see section 3.4.

## Why is the Template User “guest” Important?

By default, the uPortal framework uses the template user “guest” to manage/format the pre-authenticated portal layout. This screen displays when the user accesses the main page of the portal where the user is prompted to log in. This layout can be modified.

## Why Have Different Template Users?

### ***Customized layout***

Templates allow portal administrators to easily customize their portal for different constituent groups on campus and the portal functionality allows portal administrators to define separate layouts (tab/channel/skin configurations) for each template user type.

The following example highlights the benefit of a customized layout:

*Example: A channel layout for “students” can exist for student information (lunch menu, class registration, classifieds) and a separate channel layout can exist for “faculty” members (discussion forums, payday calendar, chat room administration). Through the templates, the Administrator can tailor the portal view for both the faculty and students. For example, the portal administrator can log in as the template user “student” and add the Groupware Channel. Now, any user associated with the “student” template will have the Groupware Channel.*

### **Templates Determine Default Group Affiliation**

Templates assist in the management of groups and permissions. In addition to determining the layout, templates decide the group’s inheritance (the default group affiliation).

The following example highlights the benefit of group inheritance, as it relates to template users:

*Example: The portal administrator wants the student to belong to the student group and the CMS - > Roles -> System -> User group (any user wanting to subscribe to an Offering in the Groupware Channel would need to belong to the CMS - > Roles -> System -> User group). In order to ensure that any user possessing the student template attribute would automatically be placed in this CMS group and inherit the permissions associated with the CMS group, the portal administrator would simply place the template user student into the CMS - > Roles -> System -> User group. Once the template user student is placed in that group, any new user belonging to the student template group would also inherit that CMS - > Roles -> System -> User group upon first log in.*

---

## **Fragments**

Template users and fragments are important mechanisms to push content to users according to group membership.

### **Understanding Fragments**

A fragment is a pre-defined tab containing at least one channel that houses content to be distributed to groups. Fragments cannot be modified by viewers. Fragments are published to users based on group affiliation. Simply stated, fragments cannot be published to templates, fragments are published to groups. There are two types of fragments: Pushed and pulled.

- Pushed fragments are not removable or editable by the user. These fragments are “pushed” to users based on group membership and merged into the layout. In fact, the delete button will always be disabled for pushed fragments since they will be added to a user’s layout no matter what. The only way to remove a pushed fragment from a user’s layout is remove that user’s permission to view the fragment.

- Pulled fragments are also pre-defined as far as design, content, and layout, but these fragments are not pushed on the user. Pulled fragments are optional in that end users have the choice whether or not to subscribe to the pulled fragment. Once the users subscribes to a pulled fragment, they cannot modify any of the channels on the subscribed fragment. Users can choose to stop receiving a pulled fragment by removing the fragment from their layout by means of user preferences.

*Important: Fragments were created to remedy the problems that can occur when users change their layout. Users who customize their layouts will break association with the template user. This means any changes to the template layout will not appear in the personalized layout. For example, if an administrator makes a change to the “Student” layout, such as adding a café menu channel, a student user with a “Student” template who had customized his layout would not see the change made to the Student template. In contrast, users mapped to the student template who have not changed their layouts would experience the layout change on the next login.*

## Benefits of Using ALM fragments

- Non-modifiable tabs based on group membership
- Availability of additional opt-in tabs based on group membership

## What Are ALM Fragments?

Academus includes an improved version of the Aggregated Layout Manager (ALM) functionality present in uPortal. ALM fragments are administrator-defined tabs containing at least one channel. ALM Fragments cannot be modified by the users who receive the fragment.

The selection and arrangement of content a user experiences in Academus portal is referred to as a layout. A given user's layout consists of applicable ALM fragments alongside personal layout content. The initial personal layout content of any user is the layout of the associated template user. The selection of ALM fragments included a user's layout is determined by their group memberships and preferences.

---

## PAGS

PAGS and template users are important as each provides a way to populate group affiliation within the portal.

### Benefits of Using PAGS

- Group association based on the most current user attributes
- Target groups for ALM fragments, such that a default layout can be composed of multiple fragments based on different group memberships
- PAGS group affiliation is not broken when users customize their layouts

### What is PAGS?

The Person Attribute Group Service (PAGS) is one group service implementation available in uPortal. Associated with each PAGS group is a rule defining which person attributes to consider and in what ways to determine whether any given person is a member of a particular group. After authentication, uPortal polls the configured user attribute sources (typically, LDAP instances and relational databases) for attributes for the authenticated user. PAGS groups are defined by logical expressions on user attributes. If the user's attributes fulfill the requirements of the logical expression associated with a particular PAGS group, the user is a member of that group.

PAGS group membership is determined "just in time" considering the latest definition of the group and latest user attributes. A users' PAGS group memberships will display to update each time the user logs into the portal as this is when user attributes are retrieved. PAGS is enabled by uncommenting a portion of code in the `compositegroupstore.xml`. From the perspective of the portal PAGS groups are read-only.

PAGS and templates user mappings both contribute to defining a users' default layout. Templates define default personal layout content while PAGS group membership can trigger the applicability of ALM fragments.

### Enabling PAGS

The following files must be modified to enable PAGS on your Academus portal:

- `/portal/unicorn/Academus/portal-tomcat-a/webapps/portal/WEB-INF/classes/properties/groups/compositeGroupServices.xml`
- `/portal/unicorn/Academus/portal-tomcat-a/webapps/portal/WEB-INF/classes/properties/groups/PAGSGroupStoreConfig.xml`

The following XML block needs to be uncommented from **compositeGroupServices.xml**:

```
<service>
  <name>pags</name>
  <service_factory>
org.jasig.portal.groups.ReferenceIndividualGroupServiceFactory
  </service_factory>
  <entity_store_factory>
org.jasig.portal.groups.pags.PersonAttributesEntityStoreFactory
  </entity_store_factory>
  <group_store_factory>
org.jasig.portal.groups.pags.PersonAttributesGroupStoreFactory
  </group_store_factory>
  <entity_searcher_factory>
org.jasig.portal.groups.pags.PersonAttributesEntitySearcherFactory
  </entity_searcher_factory>
  <internally_managed>>false</internally_managed>
  <caching_enabled>>true</caching_enabled>
</service>
```

Uncommenting this block will enable PAGS on your Academus portal. The PAGS groups must then be configured within the **PAGSGroupStoreConfig.xml** file

## Section 3. Procedures

This section outlines the procedures that relate to Portal Template Users.

---

### Creating and Configuring User Templates

The following steps detail the creation and configuration of a template user.

#### Step 1: Create the Template User

The template user can be created in either the portal database or in LDAP. The template user is created as a standard user, with a username that is descriptive of the user type that will follow the template user's layout.

To create the "alumni" template user that will define the layout for all portal users who are alumni, add a user named "alumni" to the portal database or LDAP.

*Note: Standard practice is to enter a first name of "Alumni" and a last name of "User".*

#### Step 2: Configure the Template User's Layout

Template users control the default layouts for each user type.

1. Log in to the portal as an "alumni" user.
2. Using either the Customize link, the Add Content, or Change Layout link, set up the alumni user's layout to include the desired tabs, channels, and skin.
3. When you are finished, log out.

#### Step 3: Assign the Template User to the Appropriate Group(s)

Template users control the default groups to which user groups are assigned. The Groups Manager channel is used to assign template users to the appropriate default groups.

To assign a group affiliation for the template "alumni" user:

1. Log in to the portal as an admin user.
2. Add the template user "alumni" to the group Everyone -> Students.

#### Step 4. Configure Academus to Utilize Template Users

Template users are simply portal users that have been set up for the purpose of managing user layouts. There is no database field or portal setting that designates a specific user as a template user. A user exists as a template user when there is a reference in the user attribute source to the template user name.

The first step to configure Academus to utilize template users is to designate the field within the User Attribute Source that holds (or will hold) the template user value. This field will vary based upon the User Attribute Source schema utilized by your institution.

*Example: A standard schema within LDAP is the eduPerson schema. Within the eduPerson schema, the eduPersonPrimaryAffiliation field is typically used to store a value that can be used as the template user value such as “student”, “staff” or “faculty”. In this example, the Academus portal administrator will set the following attribute in the PersonDirs.xml file:*

```
<attribute>
  <name>eduPersonPrimaryAffiliation</name>
  <alias>uPortalTemplateUserName</alias>
</attribute>
```

*The LDAP administrator will update LDAP to include the value of “alumni” in the eduPersonPrimaryAffiliation field for each portal user who is an alumni if the field is not already populated with this value.*

## Step 5: Verify Successful Completion of the Template User Setup

Using the example above:

1. Log in to the portal as an LDAP user who has a template user value (in this case, the value in “eduPersonPrimaryAffiliation”) of “alumni”, and who has never logged in before.
2. If you are presented with the tab, channel, skin, and group settings that you configured for user “alumni”, you have successfully set up this template user.

---

## Managing Template Users

It is important to understand the interaction of the portal and User Attribute Sources when managing template users.

### Retrieving Template User Values

Template User Values are retrieved each time the user logs in. When a portal user logs in, the portal queries the User Attribute Source for the value of the user attribute source attribute corresponding to the uPortalTemplateUserName portal attribute. Each time the user logs in, the portal queries all the person attribute sources for all the configured person attributes. If the uPortalTemplateUserName attribute value changes, the person will receive the new template user layout and new corresponding group associations. All previous local group affiliations will be lost.

### Visible Changes

Changes made to template user layouts will be visible to users associated with the changed template. It is expected that template user layouts will need to change over time. When a template user layout has changed, users associated with the template user will see the changes upon their next log in. This is true only in the case where users have not made manual modifications to their layout.

## Breaking Affiliation with Template Users

Users break affiliation with template users if they make changes to their layout. Once a portal user logs in and makes a layout change, such as adding or moving a channel, the user will no longer follow the layout provided for the template user. After the layout is changed, any updates made to the template user layout will not be visible to the portal users who have modified their layout.

In the case where you do not want users to make any changes to their layouts, you can lock down the preferences functionality within Academus.

## Resetting the User Layouts

In previous versions, there is no automated way to reset user layouts. The scripts included in Appendix A, “Resetting User Layouts” on page 19 provide a manual process for resetting user layouts. The baseline uPortal under DLM, as delivered in uPortal 2.5.3, provides an end-user-facing link in the preferences UI to reset the logged in user’s layout.

---

## Changing the Default Template Users

To change the default template user, identify the following property in the “portal.properties” file, and replace “demo” with the desired template user name:  
`org.jasig.portal.services.Authentication.defaultTemplateName=demo`

Once this value has been changed, the designated user template will be the default layout provided for portal users who have missing or invalid template assignments.

### “demo” Template User

By default, the “demo” template user is used in the event that a portal user does not have a template user value in their `uPortalTemplateName` person attribute or if the person attribute contains an unrecognized value. In either event, the user will receive the layout and group assignment of the template user “demo.” Demo is the default user layout designated in the uPortal framework; however, this setting can be configured.

*Note: `eduPersonPrimaryAffiliation` is the LDAP attribute this is typically mapped to this uPortal user attribute; although that is optional, configurable, and may not be the configuration of any particular Academus deployment.*

To change the default template user:

1. Identify the following line of code in the “portal.properties” file.
2. Replace “demo” with the desired template user name:  
`org.jasig.portal.services.Authentication.defaultTemplateName=demo`
3. Once this value has been changed, the designated user template will be the default layout provided for portal users who have missing or invalid template assignments.

## “guest” Template User

By default, the uPortal framework uses the “guest” user template to manage the pre-authenticated portal layout. This screen will display when accessing the main page portal page prior to log in.

To change the layout of the pre-authenticated portal page:

1. Log in as “guest”.
2. Configure the layout and skin as desired.

*Important:* After updating the guest layout, you must restart the portal in order to see changes. A restart is only required to view changes regarding the guest template user layout. This is true only for the template user guest and the layout of the pre-authenticated portal page.

## Section 4. Summary and Best Practices

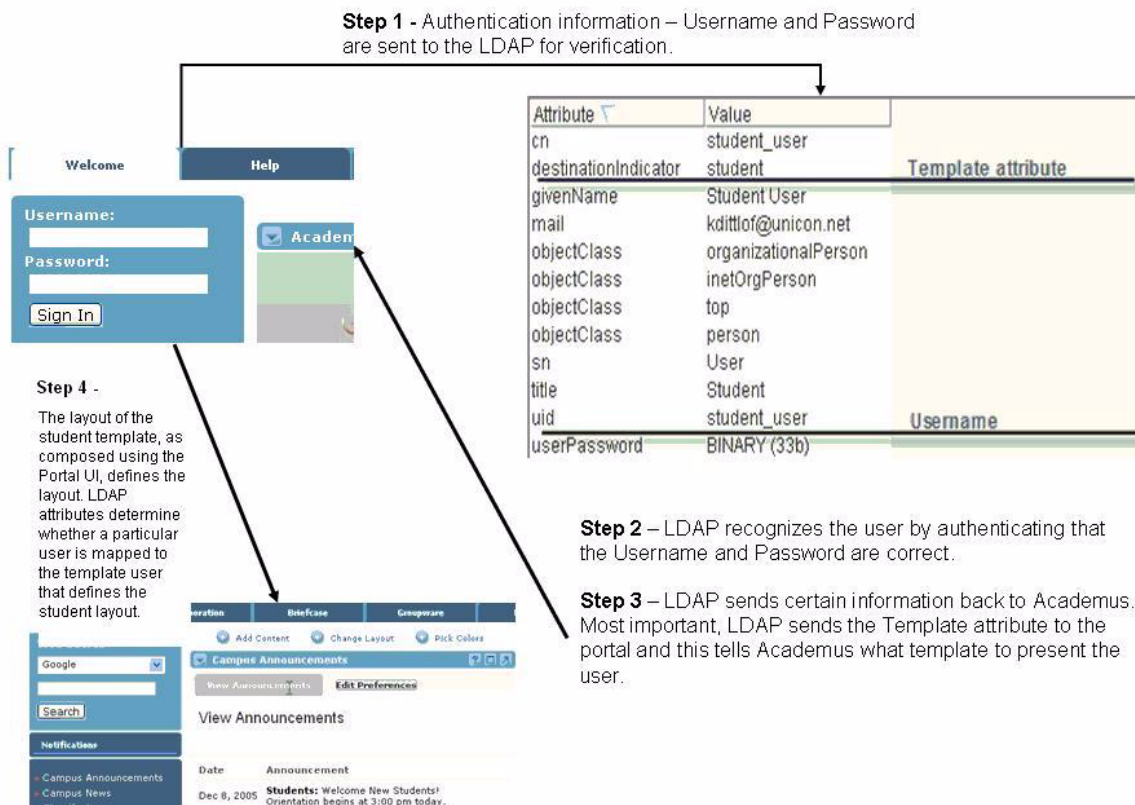
The following section provides a summary and some of the best practices that are applicable to Portal Template Users.

### Managing Academus Permissions and User Layouts

The following items provide a summary of some best practices when managing Academus Permissions and User Layouts:

- Portal login consists of authentication and authorization. During authorization, the User Attribute Source (in the following example, LDAP) sends attributes to the Academus informing the portal of values such as username, first name, and, most importantly, the template user attribute.

#### Layout Configuration using LDAP



- The template user determines both group affiliation and portal layout.
- Demo template user determines the default layout. Best practice is to log in as the demo template user in order to configure the look and feel for users that do not have a template attribute. Remember, if the uPortalTemplateUserName attribute is missing, misspelled, or simply unrecognized by Academus, the user will inherit the demo template.

- Guest template user dictates the pre-authenticated portal layout.
- Fragments supplement template layout.
- There are two types of fragments: pushed and pulled. Pushed fragments enable the portal administrator to “force” certain channels upon all users or a specific group of users.
- No User Attribute Source (for example, LDAP) - No Template!

---

## PAGS Best Practices

Use both PAGS and template users in the event that the institution leaves the Preferences on enabling the end-user’s ability to modify his layout.

- Add the PAGS Root group under Everyone in order to allow the PAGS Root to appear in the browse view of the Groups Manager channel. PAGS groups will exist in the portal depending on configuration of the Composite Group Store. Adding the PAGS Root group to Everyone is not required for the PAGS groups to work, however it is convenient to browse these groups via the Groups Manager and they will only be available via search until they are added to a local group.
- Wrap PAGS groups with local groups. For each PAGS group, create a local group and add the PAGS group as a member of that local group. For instance, it will be possible to replace the contents of the local group with an alternate PAGS group or set of PAGS groups if you later wish to change the definition of “Student”. This approach also makes it possible to add users to these groups who are not for whatever reason meeting the membership requirements of the PAGS group. Your institution may have users in complex roles, e.g. a faculty member who is also a student. Since local groups are modifiable via the portal interface, an administrator can manually add such a mutli-role user to the local groups even when the PAGS queries do not match the user's attributes.
- Then use the local group wherever you use groups to describe permissions or to push fragments.
- Store PAGS groups inside local portal groups. Using local groups as wrappers enables the admin to do the following:
- Define permissions for PAGS group via the Groups Manager
- PAGS will show up in the Groups Manager IServant channel displayed when users are targeting communications utilizing Academus tools like Notifications, Announcements, Briefcase, etc.

## Appendix A. Resetting User Layouts

As in any case where database modifications are necessary, we recommend making a database backup before running any SQL scripts.

---

### Resetting All User Layouts

```
*****
*
* File:          <reset_all_user_layouts.sql>
*
* Copyright: COPYRIGHT 2006 Unicon, Inc. All Rights Reserved
*
* This source code is the confidential and proprietary information of Unicon.
* No part of this work may be modified or used without the prior written
* consent of Unicon.
*
*****
*****
This SQL query will reset the layout for every user, with the exception
of the following users: admin, system, demo, guest, faculty, staff, and
student. You will need to copy this query and paste it into your
SQL command prompt.
*****
insert into up_user values (0,'system',NULL,NULL,NULL,NULL);

update up_user_layout
set init_struct_id = null
where user_id not in (select user_id from up_user where user_name in ('admin',
                             'system', 'demo', 'guest', 'faculty', 'staff', 'student'));

update up_user_profile
set
layout_id = null,
structure_ss_id = null,
theme_ss_id = null
where user_id not in (select user_id from up_user where user_name in ('admin',
                             'system', 'demo', 'guest', 'faculty', 'staff', 'student'));

delete from up_layout_param;
```

*Appendix A. Resetting User Layouts*  
*Resetting All User Layouts*

```
delete from up_layout_struct
where user_id not in (select user_id from up_user where user_name in ('admin',
    'system', 'demo', 'guest', 'faculty', 'staff', 'student'));
```

```
delete from up_ss_user_atts
where user_id not in (select user_id from up_user where user_name in ('admin',
    'system', 'demo', 'guest', 'faculty', 'staff', 'student'));
```

```
update up_ss_user_parm
set param_val = 'academus'
where user_id not in (select user_id from up_user where user_name in ('admin',
    'system', 'demo', 'guest', 'faculty', 'staff', 'student'))
and param_name = 'skin';
```

\*\*\*\*\*

This query is used to verify that the template user layouts are intact.

\*\*\*\*\*

```
select * from up_user where user_id not in (select user_id from up_user where
    user_name in ('admin', 'system', 'demo', 'guest', 'faculty',
    'staff', 'student'));
```

```
select * from up_user_profile where user_id not in (select user_id from up_user
    where user_name in ('admin', 'system', 'demo', 'guest', 'faculty',
    'staff', 'student'));
```

```
select * from up_user_layout where user_id not in (select user_id from up_user
    where user_name in ('admin', 'system', 'demo', 'guest', 'faculty',
    'staff', 'student'));
```

```
select * from up_layout_struct where user_id not in (select user_id from up_user
    where user_name in ('admin', 'system', 'demo', 'guest', 'faculty',
    'staff', 'student'));
```

```
select * from up_ss_user_atts where user_id not in (select user_id from up_user
    where user_name in ('admin', 'system', 'demo', 'guest', 'faculty',
    'staff', 'student'));
```

```
select * from up_ss_user_parm where user_id not in (select user_id from up_user
    where user_name in ('admin', 'system', 'demo', 'guest', 'faculty',
    'staff', 'student'));
```

---

## Resetting User Layout

```
*****
*
* File:          <reset_user_layout.sql>
*
* Copyright:    COPYRIGHT 2006 Unicon, Inc. All Rights Reserved
*
* This source code is the confidential and proprietary information of Unicon.
* No part of this work may be modified or used without the prior written
* consent of Unicon.
*
*****
****

To reset a specific user layout, replace the example user_id (1111)
with the correct user_id. Once these queries are executed against
the database, the user will be able to re-login to the system and
get the designated template user layout.

****

update up_user_layout
set init_struct_id = null
where user_id = 1111;

update up_user_profile
set
layout_id = null,
structure_ss_id = null,
theme_ss_id = null
where user_id = 1111;

delete from up_layout_param
where user_id = 1111;

delete from up_layout_struct
where user_id = 1111;

delete from up_ss_user_atts
where user_id = 1111;

****

To change a user's default template, replace the example user_id (1111)
with the correct user_id, and replace the "new template user_id" placeholder
```

*Appendix A. Resetting User Layouts*  
*Resetting User Layout*

with the desired template user\_id.

\*\*\*\*\*

```
update up_user
set user_dflt_usr_id = [new template user_id]
where user_id = 1111;
```

\*\*\*\*\*

To change a user's skin, replace the example user\_id (1111) with the user\_id of the user whose skin you want to change and replace the user\_id (2222) with the user\_id of the user who has the skin that you wish to assign to user\_id (1111).

\*\*\*\*\*

```
update up_ss_user_parm
set param_val = (
select param_val from up_ss_user_parm
where user_id = 2222 and param_name = 'skin')
where user_id = 1111
and param_name = 'skin';
```