

Sakai Permissions

Permissions in Sakai 2.1.2 are tied to Roles. A Role is a set of permissions. Roles exist within the context of a site. A user may have the access role in one site, and maintain role in another. Permissions cannot be specifically assigned to an individual user. A user is given a Role in a site, and that Role carries with it the permission set.

There are two areas where permissions affect a Sakai user – in their My Workspace, and in their roles within a various worksites. The account type controls behavior that is outside the scope of a particular site. A user's role within a site is within the scope of that site only. A user may have one account type, but may have different roles in different sites.

Appendix A shows the default roles and permissions for Sakai 2.1.2

Appendix B shows roles and permissions in use at University of Michigan.

Account Permissions

A user is added to Sakai by creating an account.¹ An account has a type (e.g., 'Instructor', or 'Student', or 'guest'), and the type can control whether the user has the ability to create new sites. The ability to create sites is outside the scope of a user created worksite. Creating new worksites is a capability accessed from a user's My Workspace Worksite Setup tool, not from within a particular worksite.

The user's account type determines which realm template the user has, and in the realm template is the control for ability to create sites. The realm template used by an account is !user.template.<type>, where type is the type of account. For example, by default Sakai has a realm template !user.template.registered. That realm has the site.add permission enabled, so any user whose account is of type 'registered' will be able to create worksites. The ability to create worksites means the user will see the New button in their My Workspace Worksite Setup tool. Without site.add, an account will not see the New button, and will therefore not be able to create new worksites.

If an account has no type, or a type for which a corresponding realm template is not found, the !user.template realm is used. The !user.template realm by default has site.add disabled, so users with a blank type cannot create worksites. The account type can be seen/changed by editing the account with the Admin's User (sakai.users) tool.

If you wanted to create a new account type - colleague - you could add a !user.template.colleague realm by editing (using the Realm tool) the !user.template.registered realm, copying it to !user.template.colleague, and then editing the !user.template.colleague realm to turn on/off the site.add permission as desired.

¹ This could happen by adding the user with the User tool (sakai.users), the New Account tool which may be on the gateway (login) page (sakai.createuser) or via a provider which supplies users from the institution's enterprise system. Some institutions are also using scripts to load users.

When an account is created using the New Account tool, it is created with type ‘registered’. By default then, these users have the ability to create new worksites. This could be changed by editing the !user.template.registered realm (Admin Realm tool) and unchecking the site.add permission.

Default Sakai realms include

!user.template	no site.add permission
!user.template.guest	no site.add permission
!user.template.registered	site.add permission enabled
!user.template.maintain	site.add permission enabled

Note: do not confuse the account type ‘maintain’ with the role of ‘maintain’ that may be given to a user in a particular site. Account type and the role in a worksite serve two different purposes.

Roles and Permissions in Worksites – site templates

The creator of a worksite is automatically a member of the worksite and by default is added as a participant when it is created. The creator is given the “maintain” role (a role specified in the site’s realm, which typically gives the site creator maintain permissions in the sit). The realm for the worksite is where roles for the worksite are defined.

Sakai can be configured so that sites of different types can have different sets of roles. Sites, like accounts, can have a type. If so, they inherit the !site.template.<type> if it exists. Otherwise, the site inherits the !site.template realm. By default, Sakai has !site.template and !site.template.course realms. Course sites inherit the !site.template.course realm, and all other types of sites (e.g., project) inherit the !site.template realm.

Since a site inherits the !site.template realm, or if the site is a course type, inherits the !site.template.course realm, the site has roles defined in the associated template. By default, roles included in !site.template are ‘maintain’ and ‘access’. Roles in the !site.template.course realm are Instructor, Student, and Teaching Assistant.

The maintain type role for !site.template and !site.template.course is ‘maintain’ and ‘Instructor’ respectively. The access type role for these two realms is ‘access’ and ‘Student’ respectively. The maintain type role allows read/write/edit throughout the site. The access type role provides read everywhere, and limited write (access can post Discussion replies, but not upload to Resources for example). These are similar to an Instructor and Student roles in a class site, or Organizer and Member roles in a project site. See Appendix A for a listing of the default permissions for the roles in !site.template and !site.template.course.

A site’s realm can be viewed/edited using the Admin’s Realm (sakai.realm) tool. A site’s realm will be named /site/<siteid>, where <siteid> is the site’s unique identifier (used in the URL of the site, e.g., 929ece94-34e3-4e71-0092-7afe89278d0e).

A typical application would be to create a !site.template.project realm, and perhaps adjust roles in the !site.project.course realm. In each, create the set of roles appropriate to those types of sites. See Appendix B for examples from UM, where course sites have an Instructor and Student role, and project sites have an Organizer and Member role. In both types of sites at UM, site creators get the ‘Owner’ role rather than ‘maintain’.

Users are added to a worksite by the site creator using Worksite Setup, and/or users may join a worksite if the worksite has been made joinable. The permissions that a user has in a worksite are determined by the user’s role in the worksite. Their assigned role is specified by the person adding the user to the worksite. If the user self-joins (using the Membership tool in their My Workspace), they are automatically assigned a role based on a role for joiner specified when the worksite was created.

Functionality to create a new role in a site is in the Admin Realm tool, and so is not available to normal users at this time. When a new role is created, permissions for the role are specified using the Realm editor.

Site and site templates included in Sakai OOTB

Site ID	Description
!admin	The Administration Workspace. Contains tools that support/admins use in configuring Sakai and troubleshooting user problems (e.g., User, Realm tools). Any account that is a member of this site has access to the admin tools, and is granted read/write/access permissions to all sites in the system.
!error	A site used to present an error page (via a sakai.iframe.site tool) when a user tries to get to a site that doesn’t exist, or for which they are not a member, or otherwise not authorized to see. Installations can tailor the message to whatever they want by editing the description of the sakai.iframe.site tool in the !error site, or pointing the tool to an html file.
!gateway	The Gateway site which users see when they go to the login page for Sakai (e.g., http://localhost:8080/portal). Installations can customize what shows up to their users on the Gateway page by editing/adding/removing tools in the !gateway site.
!urlError	Not used.
!user	The site template used when a new My Workspace is created the first time a new user logs in to Sakai. The user’s My Workspace will contain tools that are in the !user site. Institutions can customize the tools users start with in their My Workspaces by editing the !user site. Or, they can create different sets of tools for different types of user My Workspaces by creating !user.<usertype> template sites. My Workspaces are created using the !user site as a template, unless there exists a corresponding

!user.<usertype> that matches the user's account type.

!worksite An example site that can be copied with the Admin Sites tool to create new sites, but not used currently as a template.

~admin The admin's My Workspace. Contains by default the same tools as in the !admin site, but only the user account 'admin' has access to this site. Giving other accounts admin rights is done by granting other accounts access to the !admin site.

mercury

Group template realm

The new Site Group feature introduced in Sakai 2.1 introduced group templates !group.template and !group.template.course. In order for Site Groups to work properly, these templates must have the same roles defined as those in the corresponding site templates !site.template and !site.template.course. So for Sakai OOTB, !group.template has the roles maintain and access, and !site.template.course has the roles Instructor, Student, and Teaching Assistant.

The !site.helper realm

Editing a template realm affects all future sites. Editing a site's realm affects the site only. It is sometimes desirable to add a permission to all existing sites. For example, a new tool is added with its set of permissions. It would be nice to add the new tool's default permission settings to roles in all sites so that if an existing site wanted to add the new tool, permissions would be setup properly for the various roles. The !site.helper realm provides such a capability. The !site.helper realm can be used to assign permissions for a particular role in all sites.

More information on the !site.helper realm can be found in the Sakai Helper Realm doc at http://collab.sakaiproject.org/access/content/group/1097928811887-22636/Architecture%20Docs/Sakai%202/helper_realm.pdf. The document is also in subversion at /trunk/sakai/docs/architecture/sakai_helper_realm.doc

Other realms and realm templates

A few other realms are delivered with Sakai out of the box. These have special uses as follows.

!group.template	Used to specify group related permissions for tools that are group aware for sites which do not have a corresponding !group.template.<sitetype> realm.
!group.template.course	Used to specify group related permissions for tools that are group aware for course sites (sites with a type value of 'course')

!pubview	Gives permissions to non-authenticated users who are viewing public content in a site. Public content includes Announcements, Resources, Syllabus items that have been specified as being public when they were created (or subsequently edited). Public content for a site can be viewed from the Site Browser tool on the Gateway page, before a user logs in. !pubview typically has read only permissions set so that unauthenticated users can see the public content but not add/change site content.
!site.helper	Can be used to grant permission to all sites, as a way of retroactively granting permissions when a new permission is added to the system. See the document <i>Sakai Helper Realm</i> in the DG: Development (DEV) site on http://collab.sakaiproject.org , in the Resources/Architecture docs folder for more information. ²
!site.template	Realm template used for sites that do not have any type, or for which a corresponding !site.template.<sitetype> does not exist.
!site.template.course	Realm template for sites of type 'course'.
!site.template.myworkspace	Not used.
!site.user	Provides permissions to the user in their My Workspaces. When a user first logs in, their My Workspace inherits this realm by default.

² http://collab.sakaiproject.org/access/content/group/1097928811887-22636/Architecture%20Docs/Sakai%202/helper_realm.pdf

!user.template	<p>User templates are a way of granting certain permissions to users within the scope of their My Workspace based on their account type. Currently, these templates are used to grant the ability to create new sites or not via the site.new permission. If the user template has site.new checked, then the ‘New’ action appears in the user’s Worksite Setup tool in their My Workspace. If the site.new permission is not checked, then the user cannot create new worksites since the New action will be missing from their Worksite Setup toolbar.</p> <p>!user.template is the realm used for user account with no type, or accounts which have a type but for which there isn’t a corresponding !user.template.<accounttype> realm.</p>
!user.template.guest	Used for user accounts of type ‘guest’.
!user.template.maintain	Used for user accounts of type ‘maintain’
!user.template.registered	Used for user accounts of type ‘registered’
!user.template.sample	Used for user accounts of type ‘sample’

Permissions for the System Administrator

The system administrator (admin) has access to tools to which no other regular user has access. The admin has tools such as “Users,” “Realms,” and “Sites” which allow the admin to create, delete, and generally administer the associated entities. The admin also has privileges and permissions not available to any other user. For example, the admin can see and edit all sites.

When Sakai is first installed, the admin account (account name admin, pwd admin) is the only ‘pre-installed’ account, and is able to create worksites.

Other accounts can be given admin rights so that the admin account itself does not have to be used by multiple users. An account is given admin rights by editing the /site!/admin realm (the realm associated with the Administration Workspace) and using the Grant Ability function in the toolbar of the Realm editor. Add the account name in the Grant Ability page, and assign the account the admin role. That account will then see the Administration Workspace tab, and be able to access Admin tools in that worksite.

Sakai 2.1.2 default permissions

Sakai 2.1.2 includes two roles – access and maintain. The access role is a generalized role to represent a ‘normal’ user of a site, and the maintain role represents a site owner or site

administrator type role. Those with maintain can do most anything in the site. They have read/write/del everywhere, and can edit tool options and permissions. Access role users can read everywhere, write in some places, and generally cannot delete. Access users do not have site.upd permission, so they do not see the Options and Permissions items in toolbars.

Permissions for these roles are set in the realm template !site.template given to new sites. Institutions may want to change the defaults in !site.template to add roles with other permissions, or create !site.template.<type> template realms to provide custom roles and permissions to sites of particular types (e.g., course and project).

The list of permissions and settings for the default roles is shown in Appendix A.

Notes:

- Some tools do not yet have a Permissions page to control permissions from the tool (e.g., Gradebook). To set permissions, use the Admin Realm editor to edit the Realm, where you can set permissions per role in the site's Realm. A site's realm is /site/<siteid>.
- Some tools do not have permissions (e.g., Syllabus). These tools typically tie tool administrative functions to site.upd. If the role has site.upd, then users see toolbar actions associated with maintaining, administering the tool, creating new objects with the tool etc. If the role does not have site.upd, then users see a 'normal' user (student type) view of the tool and do not have edit permission in the tool.
- When a new tool is added to a release, if it has associated permissions, those new permissions are added into the default roles of both the realm template used for new sites and existing site realms. The !site.template realm will contain appropriate settings for the new tool's permissions as part of the release, but existing sites in installations will not have anything set for the new tool's permissions in their realms. .
- If you make a change to a site template realm (e.g., !site.template), those changes affect only sites created after the change. Existing sites will not be affected since their realms are already created. Some institutions have created scripts to update existing realms with new permission settings.

Announcements

↑ ANNOUNCEMENTS ?

Set permissions for Announcements in worksite "Course 200 001 W06"
(500e1942-477a-468f-80c9-28d69bde5c9)

Role	read	new	delete.any	delete.own	revise.any	revise.own	read.drafts	all.groups
Instructor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Student	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teaching Assistant	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Permission	!site.template entry	Description
new	annc.new	Allows users to create new announcements.
read	annc.read	Allows users to read announcements. This permission is needed to view the list of announcements.
revise.any	annc.revise.any	Allows users to edit any announcement, created by anyone.
revise.own	annc.revise.own	Allows user to edit only announcements they created.
delete.any	annc.delete.any	User can delete any announcement regardless of who created it.
delete.own	annc.delete.own	User can delete only announcements they created.
read.drafts	annc.read.drafts	Ability to read draft announcements made by others. Users can always view draft announcements they create.
all.groups	annc.all.groups	Maintain type role members of the site (Instructors and the like) might expect to have permissions to see and manipulate the announcements in the site as well as all the site groups, without having explicit membership in each group. If the user's membership in the site includes annc.all.groups, then the user has access to all the groups in the site without needing explicit group membership.

Assignments

Role	read	new	revise	delete	submit	grade
access	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
maintain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Save Cancel

Permission	!site.template entry	Description
read	asn.read	Allows user to view the assignment list. The list they see will differ depending on the <code>asn.new</code> permission. Those with <code>asn.new</code> see an 'instructor' list (list of all assignments, including drafts, with action links for revising, deleting). Those without <code>asn.new</code> see a 'student' list (list of open assignments, where clicking on the assignment opens it for submission).
new	asn.new	Gives user the ability to create new assignments. Also controls the type of assignment list view the user sees (see <code>asn.read</code> above).
revise	asn.revise	Gives user the ability to revise assignments.
delete	asn.delete	Gives user the ability to delete assignments.
submit	asn.submit	Gives user the ability to submit assignments. This includes 'instructor' type roles, which can submit as though they were a 'student' using a link on the Student View page. Without this permission, the 'Submit as Student' function does not appear to roles with <code>asn.new</code> .
grade	asn.grade	Gives the role the ability to grade assignments. See note below.

Notes: If the role has `asn.new` permission, then that role also has the ability to grade (any role with `asn.new` also has the ability to grade, regardless of the `asn.grade` setting). If the role does not have `asn.new`, then grading permission can be granted by giving the role `asn.grade`.

asn.submit must be granted to allow submissions regardless of any other permission setting. So an Instructor type role should have asn.submit to allow the instructor to make submissions using the 'Submit as student' action when in the Student View display.

asn.submit and asn.grade are not currently set on in the maintain role as of Sakai 2.0.1. Jira SAK-1152 addresses this bug.

Chat Room

↑ CHAT ROOM ?

Set permissions for Chat room in worksite "Course 200 001 W06"
(500e1942-477a-468f-80c9-28d69bde5c9)

Role	read	new	delete.any	delete.own
Instructor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Student	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teaching Assistant	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Permission	!site.template entry	Description
new	chat.new	Gives user the ability to post new messages. Required in order to view the chat message type in field.
read	chat.read	Gives user the ability to read chat messages. Required in order to view the chat message window.
delete.any	chat.delete.any	Gives the user the ability to delete any chat message, posted by any user.
delete.own	chat.delete.own	Not currently implemented.

Discussion

↑ DISCUSSION ?

Set permissions for Discussion in worksite "Course 200 001 W06"
(500e1942-477a-468f-80c9-28d69bde5c9)

Role	read	new	new.topic	delete.any	delete.own	revise.any	revise.own
Instructor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Student	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Teaching Assistant	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

Permission	!site.template entry	Description
new	disc.new	Required in order to post any kind of reply - either to a topic or to another reply.
read	disc.read	Gives the user the ability to view Discussion content, including the list of Categories, Topics, and Replies.
revise.any	disc.revise.any	Not currently implemented. There is no UI for revising discussion messages.
revise.own	disc.revise.own	Not currently implemented. There is no UI for revising discussion messages.
delete.any	disc.delete.any	Gives user the ability to delete discussion categories, topics or replies posted by any user. Required in order to delete. Deleting a Category or Topic deletes all replies to the Topic as well.
delete.own	disc.delete.own	Gives user the ability to delete discussion replies that they have posted. Note that currently there is no way to grant the ability to delete just the Topics and/or Categories that a user has created.
new.topic	disc.new.topic	Gives the user the ability to create new Categories and Topics. But note that since creating a topic includes creating a message, that disc.new is also required.

Note: Control over the type of replies allowed in a particular discussion topic is not controlled by a permission setting. When a topic is created, Topic Format options allow control over how replies to the topic are handled. The options are:

Within this topic, allow replies to any message

Within this topic, only allow replies directly to this message

Depending on the option, the discussion replies will appear in the Discussion list as follows:

Category

Topic 1 (set to allow replies to topic only)

Reply 1
Reply 2
Reply A
Reply a
Reply 6
Reply D

Topic 2 (set to allow replies to messages or topic)

Reply 1
Reply 2
Reply A
Reply a
Reply 6
Reply D

Dropbox

<u>!site.template entry</u>	<u>Description</u>
dropbox.maintain	Roles granted dropbox.maintain will be have read/write access to all dropboxes. They will see the entire list of dropboxes and can navigate into any dropbox, and can read and upload files there.
dropbox.own	Roles granted dropbox.own will have an individual dropbox and will not be able to see any other user's dropbox.

The dropbox permissions are editable only using the Admin Realm editor since there is no Permission page for the Dropbox tool. By design, there shouldn't be a Permission page available from the Dropbox tool. The dropbox.maintain permission can cause severe privacy problems if misused, so up to this point control of that permission hasn't been exposed to users.

Email Archive

↑ EMAIL ARCHIVE ?

Set permissions for Email Archive in worksite "Project One"
(4f7b94c5-8d05-4556-8005-407cb49fe870)

Role	read	new	delete.any
access	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
maintain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

Permission	!site.template entry	Description
new	mail.new	Gives user the ability to send email to the site, which is stored in the email archive and sent on to site participants with read permission.
read	mail.read	Gives user the ability to view the email archive list and read emails in the list.
delete.any	mail.delete.any	Allows user to delete any email message in the archive, sent by any user.

Message Center

Message Center is not part of the 2.1.2 release, so these permissions will not show in a site's realm unless you add the Message Center tool to your build.

<u>!site.template entry</u>	<u>Description</u>
messagecenter.changeSettings	Allows users to change the default template settings in the discussion forum area.
messagecenter.deleteAny	Allows user to delete any discussion topic posting.
messagecenter.deleteOwn	Allows user to delete their own discussion topic postings.
messagecenter.markAsRead	Allows user to mark discussion topic postings as read.
messagecenter.movePostings	Allows user to move discussion topic postings to another topic.
messagecenter.newForum	Allows user to create a new forum.
messagecenter.newResponse	Allows user to respond to a discussion topic.
messagecenter.newResponseToResponse	Allows user to respond to a response to a discussion topic.
messagecenter.newTopic	Allows user to create a new topic.
messagecenter.postToGradebook	Allows user to grade a discussion topic posting.
messagecenter.read	Allows user to read a discussion topic posting.
messagecenter.reviseAny	Allows user to revise any discussion topic posting.
messagecenter.reviseOwn	Allows user to revise their own discussion topic postings only.

Resources

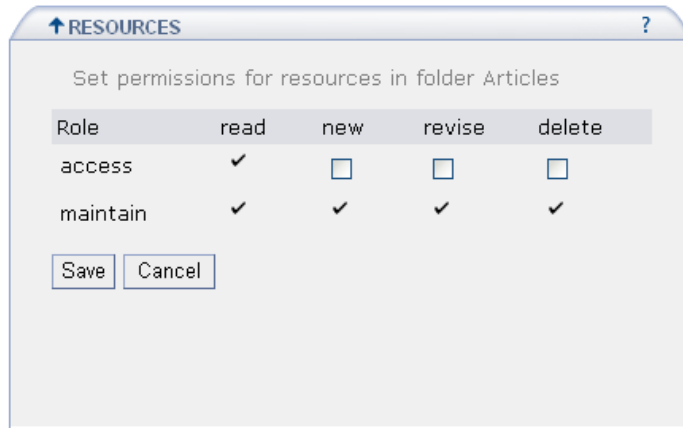
At the top level, when you first enter Resources before navigating to any sub-folder, the Permissions control item in the toolbar is used to set permissions for the top level 'root' folder of the site's Resources. These permissions are inherited by all sub-folders.

Permissions granted at the top level cannot be removed in sub-folders. There currently is no 'deny' capability. That means that you cannot at this time prevent a role from reading any subfolder (since to even get to the sub-folder, you would have to give the role 'read' permissions at the top level, and once given there, the permission can't be taken away in a sub-folder).

Additional permissions can be granted in sub-folders. You can add to sub-folder permissions, but not take away permissions that have been granted in a parent folder.

Role	read	new	revise	delete
access	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
maintain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Resource permission settings which control Role capabilities in the Resources tool at the top level 'root' folder.



Resource permission settings for a sub-folder. Note that some settings are not changeable since the sub-folder inherits the top level folder permissions, and a permission granted in a parent folder cannot be removed in the sibling.

Permission	!site.template entry	Description
read	content.read	Allows users to access the Resources area- view the list of Resources and navigate to sub-folders. There is currently no way to remove read in a sibling folder when it has been granted to the parent folder. read is needed in order to see toolbar actions and anything in the resources list.
new	content.new	Allows user to upload files to Resources, or create new text, html, or URL resources.
revise	content.revise	Allows user to revise files in Resources created by any user. revise is needed along with delete in order to delete files.
delete	content.delete	Allows user to delete files in Resources created by any user. revise is also needed in order to be able to delete.

Schedule

↑ SCHEDULE ?

Set permissions for Schedule in worksite "jlproj2" (1c661bea-e929-4c72-803e-65b17bb6c8ae)

Role	read	new	delete	import	revise
access	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
maintain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Permission	!site.template entry	Description
read	calendar.read	Allows users to view the schedule and to read schedule items. This permission is needed in order to see schedule events.
new	calendar.new	Allows users to create a new schedule item.
delete	calendar.delete	Allows users to delete schedule events. Currently, delete alone does nothing. revise is needed in order to delete. SAK-144
import	calendar.import	Allows users to import schedule events from Outlook, Meeting Maker, or csv files. (Import adds events without syncing – importing the same file multiple times results in multiple copies of events)
revise	calendar.revise	Allows users to revise (and currently, delete) events.

Tests and Quizzes

<u>!site.template entry</u>	<u>Description</u>
assessment.createAssessment	Allows user to create a new assessment.
assessment.deleteAssessment.any	User can delete any assessment created by anyone.
assessment.deleteAssessment.own	User can delete assessments they created.
assessment.editAssessment.any	User can edit any assessment created by anyone.
assessment.editAssessment.own	User can edit assessments they created.
assessment.gradeAssessment.any	User can grade any assessment.
assessment.gradeAssessment.own	User can grade assessments they created.
assessment.publishAssessment.any	User can publish any assessment.
assessment.publishAssessment.own	User can publish only assessments they have created
assessment.questionpool.copy.own	Users can make a copy of any assessment pool.
assessment.questionpool.create	User can create a question pool.
assessment.questionpool.delete.own	User can delete their own question pools.
assessment.questionpool.edit.own	User can edit question pools they have created.
assessment.submitAssessmentForGrade	User has the ability to create a submission to assessments which will be available to the grader for evaluation.
assessment.takeAssessment	User has the ability to take an assessment.
assessment.template.create	User has the ability to create a new template that can then be used to control assessment settings.

assessment.template.delete.own

User can delete templates they have created.

assessment.template.edit.own

User can edit templates they have created.

Other permissions that are not accessible via a tool Permission page can be set using the Admin Realm tool. Edits to particular roles should be made in the !site.template realms.

alias.add alias.del alias.upd	From an early implementation that allowed setting a site alias
annnc.all.groups annnc.delete.any annnc.delete.own annnc.new annnc.read annnc.revise.any annnc.revise.own	Announcement tool permissions
asn.delete asn.grade asn.new asn.read asn.revise asn.submit	Assignment tool permissions
calendar.delete calendar.import calendar.new calendar.read calendar.revise	Schedule tool permissions
chat.delete.any chat.delete.own chat.new chat.read	Chat tool permissions
content.delete content.new content.read content.revise	Resource tool permissions
disc.delete.any disc.delete.own disc.new disc.new.topic disc.read disc.read.drafts disc.revise.any disc.revise.own	Discussion tool permissions
dropbox.maintain dropbox.own	Dropbox tool permissions
gradebook.access gradebook.maintain	Gradebook tool permissions
mail.delete.any mail.new mail.read	Email Archive tool permissions
prefs.add prefs.del prefs.upd	Preference permissions (Not implemented as of Sakai 2.0)
realm.add realm.del realm.upd realm.upd.own	used in !user.template used in !site.template used in !site.template used in !user.template In !user.template, the .auth role has realm.add and realm.upd.own. In !site.template, the maintain role has realm.del and realm.upd
site.add	Allows the user to create sites (they see the New button in their My Workspace. This permission is set in the !user.template realms, not applicable in !site.template realms

site.add.usersite	Used in the !user.template – gives user the right to create their my workspace (not applicable in the !site.template realm)
site.del	Gives user the right to delete the site.
site.upd	Gives user the right to edit the site – users will see the Options and Permissions actions in Toolbars
site.upd.grp.mbrshp	For the Section Info tool, for roles that do not have site.upd, allows user to modify (but not create) membership in sections.
site.upd.site.mbrshp	For the Section Info tool, for roles that do not have site.upd, allows user to add and remove participants from the site.
site.viewRoster	Allows users to view the roster in Site Info. This is settable only via Admin Realms tool when editing the !site.template (e.g., there is not Permissions page in Site Info)
site.visit	Gives user the ability to see the site tab. Without this, even though a user may be a participant of the site, they will not see the tab.
site.visit.unp	Gives user the ability to see the site when it is unpublished.
user.add	Used in the !user.template realm. Allows a user to create a new account. Could be set in the .anon and .auth role to let anyone create an account.
user.del	
user.upd.any	
user.upd.own	Allows a user to revise their own user information. Used in the !user.template realm.
	In the default !user.template, the .auth role has user.add, user.upd.own. The .anon role has user.add

Permissions added with the Message Center tool

messagecenter.changeSettings	
messagecenter.deleteAny	
messagecenter.deleteOwn	
messagecenter.markAsRead	
messagecenter.movePostings	
messagecenter.newForum	Message Center tool permissions
messagecenter.newResponse	
messagecenter.newResponseToResponse	
messagecenter.newTopic	
messagecenter.postToGradebook	
messagecenter.read	
messagecenter.reviseAny	
messagecenter.reviseOwn	

Appendix A

Default roles and permissions supplied in !site.template, !site.template.course in Sakai 2.1.2.

!site.template is the template realm that worksites inherit when they are created. If a corresponding template realm exists to match the site type, then that template realm is used instead. For example, if a !site.template.course realm exists, sites of type 'course' are given that realm.

2.1.2	!site.template		!site.template.course		
	access	maintain	Student	Teaching Assistant	Instructor
alias.add					
alias.del					
alias.upd					
annc.all.groups		✓			✓
annc.delete.any		✓			✓
annc.delete.own		✓			✓
annc.new		✓			✓
annc.read	✓	✓	✓	✓	✓
annc.read.drafts		✓			✓
annc.revise.any		✓			✓
annc.revise.own		✓			✓
asn.delete		✓			✓
asn.grade		✓			✓
asn.new		✓			✓
asn.read	✓	✓	✓	✓	✓
asn.revise		✓			✓
asn.submit	✓	✓	✓	✓	✓
assessment.createAssessment		✓			✓
assessment.deleteAssessment.any		✓			✓
assessment.deleteAssessment.own		✓			✓
assessment.editAssessment.any		✓			✓
assessment.editAssessment.own		✓			✓
assessment.gradeAssessment.any		✓		✓	✓
assessment.gradeAssessment.own		✓		✓	✓
assessment.publishAssessment.any		✓			✓
assessment.publishAssessment.own		✓			✓
assessment.questionpool.copy.own		✓			✓
assessment.questionpool.create		✓			✓
assessment.questionpool.delete.own		✓			✓
assessment.questionpool.edit.own		✓			✓
assessment.submitAssessmentForGrade	✓		✓		
assessment.takeAssessment	✓		✓		
assessment.template.create		✓			✓
assessment.template.delete.own		✓			✓
assessment.template.edit.own		✓			✓

calendar.delete		✓			✓
calendar.import					
calendar.new		✓			✓
calendar.read	✓	✓	✓	✓	✓
calendar.revise		✓			✓
chat.delete.any		✓			✓
chat.delete.own		✓			✓
chat.new	✓	✓	✓	✓	✓
chat.read	✓	✓	✓	✓	✓
content.delete		✓			✓
content.new		✓			✓
content.read	✓	✓	✓	✓	✓
content.revise		✓			✓
crud.create					
crud.delete					
crud.read					
crud.update					
dis.dis.add					
dis.dis.del					
dis.dis.read					
dis.dis.upd					
dis.grp.add					
dis.grp.del					
dis.grp.read					
dis.grp.upd					
dis.info.add					
dis.info.del					
dis.info.read					
dis.info.upd					
dis.path.add					
dis.path.del					
dis.path.read					
dis.path.upd					
dis.path.upd.comm					
dis.status.add					
dis.status.del					
dis.status.read					
dis.status.upd					
dis.step.add					
dis.step.del					
dis.step.read					
dis.step.upd					
disc.delete.any		✓			✓
disc.delete.own		✓			✓
disc.new	✓	✓	✓	✓	✓
disc.new.topic		✓			✓
disc.read	✓	✓	✓	✓	✓
disc.revise.any		✓			✓

disc.revise.own	✓	✓	✓	✓	✓
dropbox.maintain		✓			✓
dropbox.own	✓		✓	✓	
gradebook.editAssignments		✓			✓
gradebook.gradeAll		✓			✓
gradebook.gradeSection				✓	
gradebook.viewOwnGrades	✓		✓	✓	
mail.delete.any		✓			✓
mail.new		✓			✓
mail.read	✓	✓	✓	✓	✓
melete.author					✓
melete.student					
metaobj.create		✓			✓
metaobj.edit		✓			✓
metaobj.export					
metaobj.publish		✓			✓
metaobj.suggest.global.publish		✓			✓
prefs.add					
prefs.del					
prefs.upd					
realm.add					
realm.del		✓			✓
realm.upd		✓			✓
realm.upd.own					
rwiki.admin		✓			✓
rwiki.create	✓	✓	✓	✓	✓
rwiki.read	✓	✓	✓	✓	
rwiki.superadmin					
rwiki.update	✓	✓	✓	✓	
section.role.instructor		✓			
section.role.student	✓		✓		
section.role.ta				✓	
site.add					
site.add.usersite					
site.del		✓			✓
site.upd		✓			✓
site.upd.grp.mbrshp				✓	
site.upd.site.mbrshp					
site.viewRoster					
site.visit	✓	✓	✓	✓	✓
site.visit.unp		✓			✓
user.add					
user.del					
user.upd.any					
user.upd.own					

Permissions added if you add the Message Center tool to your build.

messagecenter.changeSettings					✓
------------------------------	--	--	--	--	---

messagecenter.deleteAny					✓
messagecenter.deleteOwn					
messagecenter.markAsRead			✓		✓
messagecenter.movePostings					✓
messagecenter.newForum					✓
messagecenter.newResponse			✓		
messagecenter.newResponseToResponse			✓		✓
messagecenter.newTopic					✓
messagecenter.postToGradebook					✓
messagecenter.read			✓		✓
messagecenter.reviseAny					✓
messagecenter.reviseOwn					✓

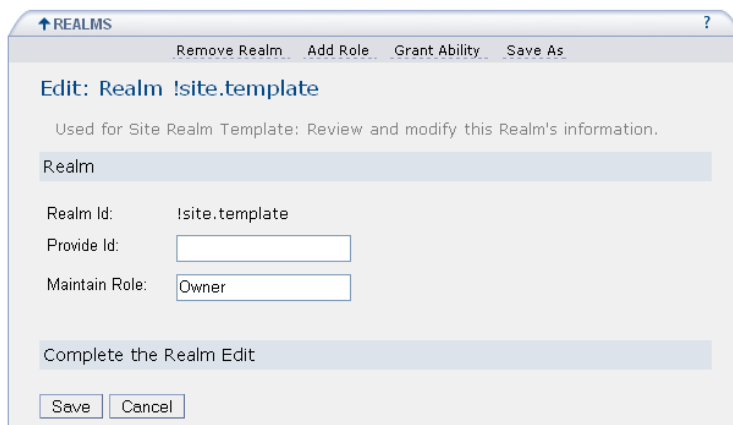
Appendix B

University of Michigan Roles and Permissions.

Site templates:

- | | |
|------------------------|--|
| !site.template | Default if no matching site type |
| !site.template.course | For course sites, contains roles of Owner, Instructor, Affiliate, Assistant, Student, Observer |
| !site.template.project | For project sites, contains roles of Owner, Administrator, Member, Observer |

At UM, we do not use the default roles of access and maintain (except for some sites created early in our pilot phase) !site.templates have the 'Maintain Role' field set to Owner so that creators get the role of 'Owner'. The Maintain Role field is visible when editing the realm using the Admin Realm editor.



The screenshot shows a web interface for editing a realm. At the top, there is a navigation bar with the title '↑ REALMS' and a help icon '?'. Below this is a menu with options: 'Remove Realm', 'Add Role', 'Grant Ability', and 'Save As'. The main heading is 'Edit: Realm !site.template'. A sub-heading reads 'Used for Site Realm Template: Review and modify this Realm's information.' Below this is a section titled 'Realm' with a light blue background. It contains three fields: 'Realm Id:' with the value '!site.template', 'Provide Id:' with an empty text box, and 'Maintain Role:' with the value 'Owner'. Below the 'Realm' section is another light blue section titled 'Complete the Realm Edit' which contains two buttons: 'Save' and 'Cancel'.

UM permissions for roles used in course sites

The following roles and associated permissions are defined in the !site.template.course realm from University of Michigan's 2.0 installation

!site.template.course

2.1.2	Affiliate	Assistant	Instructor	Observer	Owner	Student
alias.add						
alias.del						
alias.upd						
annc.all.groups	✓	✓	✓		✓	
annc.delete.any	✓		✓		✓	
annc.delete.own	✓	✓	✓		✓	
annc.new	✓	✓	✓		✓	
annc.read		✓	✓	✓	✓	✓
annc.read.drafts	✓	✓	✓		✓	
annc.revise.any	✓	✓	✓		✓	
annc.revise.own	✓	✓	✓		✓	
asn.delete	✓		✓		✓	
asn.grade	✓	✓	✓		✓	
asn.new	✓	✓	✓		✓	
asn.read	✓	✓	✓	✓	✓	✓
asn.revise	✓	✓	✓		✓	
asn.submit	✓	✓	✓		✓	✓
assessment.createAssessment	✓	✓	✓		✓	
assessment.deleteAssessment.any	✓	✓	✓		✓	
assessment.deleteAssessment.own	✓	✓	✓		✓	
assessment.editAssessment.any	✓	✓	✓		✓	
assessment.editAssessment.own	✓	✓	✓		✓	
assessment.gradeAssessment.any	✓	✓	✓		✓	
assessment.gradeAssessment.own	✓	✓	✓		✓	
assessment.publishAssessment.any	✓	✓	✓		✓	
assessment.publishAssessment.own	✓	✓	✓		✓	
assessment.questionpool.copy.own	✓	✓	✓		✓	
assessment.questionpool.create	✓	✓	✓		✓	
assessment.questionpool.delete.own	✓	✓	✓		✓	
assessment.questionpool.edit.own	✓	✓	✓		✓	
assessment.submitAssessmentForGrade						✓
assessment.takeAssessment						✓
assessment.template.create	✓	✓	✓		✓	
assessment.template.delete.own	✓	✓	✓		✓	
assessment.template.edit.own	✓	✓	✓		✓	
calendar.delete	✓	✓	✓		✓	
calendar.import						
calendar.new	✓	✓	✓		✓	
calendar.read	✓	✓	✓	✓	✓	✓
calendar.revise	✓	✓	✓		✓	

chat.delete.any	✓		✓		✓	
chat.delete.own	✓		✓		✓	
chat.new	✓	✓	✓		✓	✓
chat.read	✓	✓	✓	✓	✓	✓
content.delete	✓	✓	✓		✓	
content.new	✓	✓	✓		✓	
content.read	✓	✓	✓	✓	✓	✓
content.revise	✓	✓	✓		✓	
crud.create						
crud.delete						
crud.read						
crud.update						
dis.dis.add	✓		✓		✓	
dis.dis.del	✓		✓		✓	
dis.dis.read	✓		✓	✓	✓	✓
dis.dis.upd	✓		✓		✓	
dis.grp.add	✓		✓			
dis.grp.del	✓		✓			
dis.grp.read	✓		✓			
dis.grp.upd	✓		✓			
dis.info.add						
dis.info.del						
dis.info.read						
dis.info.upd						
dis.path.add					✓	✓
dis.path.del					✓	✓
dis.path.read	✓	✓	✓	✓	✓	✓
dis.path.upd	✓	✓	✓			✓
dis.path.upd.comm						
dis.status.add	✓	✓	✓		✓	
dis.status.del	✓	✓	✓		✓	
dis.status.read	✓	✓	✓		✓	✓
dis.status.upd	✓		✓		✓	
dis.step.add	✓		✓		✓	
dis.step.del	✓		✓		✓	
dis.step.read	✓	✓	✓	✓	✓	
dis.step.upd	✓		✓		✓	
disc.delete.any	✓		✓		✓	
disc.delete.own	✓		✓		✓	
disc.new	✓	✓	✓		✓	✓
disc.new.topic	✓	✓	✓		✓	
disc.read	✓	✓	✓	✓	✓	✓
disc.revise.any	✓		✓		✓	
disc.revise.own	✓	✓	✓		✓	✓
dropbox.maintain	✓	✓	✓		✓	
dropbox.own						✓
gradebook.editAssignments	✓	✓	✓		✓	
gradebook.gradeAll	✓		✓		✓	

gradebook.gradeSection		X				
gradebook.viewOwnGrades						✓
mail.delete.any	✓		✓		✓	
mail.new	✓	✓	✓		✓	
mail.read		✓	✓	✓	✓	✓
melete.author	✓	✓	✓		✓	
melete.student				✓		✓
metaobj.create	✓	✓	✓		✓	
metaobj.edit	✓	✓	✓		✓	
metaobj.export						
metaobj.publish	✓	✓	✓		✓	
metaobj.suggest.global.publish	✓		✓		✓	
prefs.add						
prefs.del						
prefs.upd						
realm.add						
realm.del	✓	✓	✓		✓	
realm.upd	✓	✓	✓		✓	
realm.upd.own						
rwiki.admin	✓	✓	✓		✓	
rwiki.create	✓	✓	✓		✓	
rwiki.read	✓	✓	✓	✓	✓	✓
rwiki.superadmin						
rwiki.update	✓	✓	✓		✓	
section.role.instructor	✓		✓		✓	
section.role.student						✓
section.role.ta		✓				
site.add						
site.add.usersite						
site.del					✓	
site.upd	✓	✓	✓		✓	
site.upd.grp.mbrshp						
site.upd.site.mbrshp						
site.viewRoster	✓	✓	✓		✓	
site.visit	✓	✓	✓	✓	✓	✓
site.visit.unp	✓	✓	✓		✓	
user.add						
user.del						
user.upd.any						
user.upd.own						

Comments:

We have mail.read disabled for the Affiliate role so that they don't receive emails. Affiliates are in many sites and were overwhelmed with email. Unchecking mail.read prevents them from seeing email in the site, and so they do not get email forwarded from the site. If they need to check email in a particular site for some reason, they can go to the site and add the mail.read permission to the Affiliate role in the Email Archive tool Permission page.

The Instructor role differs from Owner role only in that Owner has site.del, but Instructor does not.

UM permissions for roles used in project sites

The following roles and associated permissions are defined in the !site.template.project realm.

!site.template.project

2.1.2	Member	Observer	Organizer	Owner
alias.add				
alias.del				
alias.upd				
annc.all_groups			✓	✓
annc.delete.any			✓	✓
annc.delete.own	✓		✓	✓
annc.new	✓		✓	✓
annc.read	✓	✓	✓	✓
annc.read.drafts			✓	✓
annc.revise.any			✓	✓
annc.revise.own	✓		✓	✓
asn.delete			✓	✓
asn.grade			✓	✓
asn.new			✓	✓
asn.read	✓	✓	✓	✓
asn.revise			✓	✓
asn.submit	✓		✓	✓
assessment.createAssessment			✓	✓
assessment.deleteAssessment.any			✓	✓
assessment.deleteAssessment.own			✓	✓
assessment.editAssessment.any			✓	✓
assessment.editAssessment.own			✓	✓
assessment.gradeAssessment.any			✓	✓
assessment.gradeAssessment.own			✓	✓
assessment.publishAssessment.any			✓	✓
assessment.publishAssessment.own			✓	✓
assessment.questionpool.copy.own			✓	✓
assessment.questionpool.create			✓	✓
assessment.questionpool.delete.own			✓	✓
assessment.questionpool.edit.own			✓	✓
assessment.submitAssessmentForGrade	✓			
assessment.takeAssessment	✓			
assessment.template.create			✓	✓
assessment.template.delete.own			✓	✓
assessment.template.edit.own			✓	✓
calendar.delete	✓		✓	✓
calendar.import				

calendar.new	✓		✓	✓
calendar.read	✓	✓	✓	✓
calendar.revise	✓		✓	✓
chat.delete.any				✓
chat.delete.own				✓
chat.new	✓		✓	✓
chat.read	✓	✓	✓	✓
content.delete	✓		✓	✓
content.new	✓		✓	✓
content.read	✓	✓	✓	✓
content.revise	✓		✓	✓
crud.create				
crud.delete				
crud.read				
crud.update				
dis.dis.add				✓
dis.dis.del				✓
dis.dis.read		✓		✓
dis.dis.upd				✓
dis.grp.add				
dis.grp.del				
dis.grp.read				
dis.grp.upd				
dis.info.add				
dis.info.del				
dis.info.read				
dis.info.upd				
dis.path.add				✓
dis.path.del				✓
dis.path.read	✓	✓	✓	✓
dis.path.upd	✓		✓	
dis.path.upd.comm				
dis.status.add	✓		✓	✓
dis.status.del	✓		✓	✓
dis.status.read	✓		✓	✓
dis.status.upd	✓			✓
dis.step.add				✓
dis.step.del				✓
dis.step.read		✓	✓	✓
dis.step.upd				✓
disc.delete.any			✓	✓
disc.delete.own			✓	✓
disc.new	✓		✓	✓
disc.new.topic	✓		✓	✓
disc.read	✓	✓	✓	✓
disc.revise.any			✓	✓
disc.revise.own	✓		✓	✓
dropbox.maintain			✓	✓

dropbox.own	✓			
gradebook.editAssignments			✓	✓
gradebook.gradeAll			✓	✓
gradebook.gradeSection				
gradebook.viewOwnGrades	✓			
mail.delete.any			✓	✓
mail.new	✓		✓	✓
mail.read	✓	✓	✓	✓
melete.author			✓	✓
melete.student	✓	✓		
metaobj.create			✓	✓
metaobj.edit			✓	✓
metaobj.export				
metaobj.publish			✓	✓
metaobj.suggest.global.publish			✓	✓
prefs.add				
prefs.del				
prefs.upd				
realm.add				
realm.del	✓		✓	✓
realm.upd	✓		✓	✓
realm.upd.own				
rwiki.admin			✓	✓
rwiki.create	✓		✓	✓
rwiki.read	✓	✓	✓	✓
rwiki.superadmin				
rwiki.update	✓		✓	✓
section.role.instructor			✓	✓
section.role.student	✓			
section.role.ta				
site.add				
site.add.usersite				
site.del				✓
site.upd			✓	✓
site.upd.grp.mbrshp				
site.upd.site.mbrshp				
site.viewRoster	✓		✓	✓
site.visit	✓	✓	✓	✓
site.visit.unp			✓	✓
user.add				
user.del				
user.upd.any				
user.upd.own				