

---+ Package TWiki::Users::TWikiUserMapping

The User mapping is the process by which TWiki maps from a username (a login name) to a wikiname and back. It is also where groups are defined.

By default TWiki maintains user topics and group topics in the Main that define users and group. These topics are

- TWikiUsers - stores a mapping from usernames to TWiki names
- WikiName - for each user, stores info about the user
- GroupNameGroup - for each group, a topic ending with "Group" stores a list of users who are part of that group.

Many sites will want to override this behaviour, for example to get users and groups from a corporate database.

This class implements the basic TWiki behaviour using topics to store users, but is also designed to be subclassed so that other services can be used.

Subclasses should be named 'XxxxUserMapping' so that configure can find them.

ClassMethod new (\$session, \$impl)

Constructs a new user mapping handler of this type, referring to \$session for any required TWiki services.

ObjectMethod finish ()

Break circular references.

ObjectMethod supportsRegistration () -> false

return 1 if the UserMapper supports registration (ie can create new users)

ObjectMethod handlesUser (\$cUID, \$login, \$wikiname) -> \$boolean

Called by the TWiki::Users object to determine which loaded mapping to use for a given user.

The user can be identified by any of \$cUID, \$login or \$wikiname. Any of these parameters may be undef, and they should be tested in order; cUID first, then login, then wikiname. This mapping is special - for backwards compatibility, it assumes responsibility for *all* non BaseMapping users. If you're needing to mix the TWikiuserMapping with other mappings, define \$this->{mapping_id} = 'TWikiUserMapping_';

ObjectMethod login2cUID (\$login, \$dontcheck) -> \$cUID

Convert a login name to the corresponding canonical user name. The canonical name can be any string of 7-bit alphanumeric and underscore characters, and must correspond 1:1 to the login name. (undef on failure)

(if dontcheck is true, return a cUID for a nonexistant user too. This is used for registration)

ObjectMethod getLoginName (\$cUID) -> login

Converts an internal cUID to that user's login (undef on failure)

ObjectMethod addUser

(\$login, \$wikiname, \$password, \$emails, \$mcp) -> \$cUID

throws an Error::Simple

Add a user to the persistent mapping that maps from usernames to wikinames and vice-versa. The default implementation uses a special topic called "TWikiUsers" in the users web. Subclasses will provide other implementations (usually stubs if they have other ways of mapping usernames to wikinames). Names must be acceptable to \$TWiki::cfg{NameFilter} \$login must **always** be specified. \$wikiname may be undef, in which case the user mapper should make one up. This function must return a **canonical user id** that it uses to uniquely identify the user. This can be the login name, or the wikiname if they are all guaranteed unique, or some other string consisting only of 7-bit alphanumeric characters and underscores. If you fail to create a new user (for eg your Mapper has read only access), throw Error::Simple('Failed to add user: '.\$ph->error());

ObjectMethod removeUser (\$cUID) -> \$boolean

Delete the user's entry. Removes the user from the password manager and user mapping manager. Does **not** remove their personal topics, which may still be linked.

ObjectMethod getWikiName (\$cUID) -> \$wikiname

Map a canonical user name to a wikiname. If it fails to find a WikiName, it will attempt to find a matching loginname, and use an escaped version of that. If there is no matching WikiName or LoginName, it returns undef.

ObjectMethod userExists (\$cUID) -> \$boolean

Determine if the user already exists or not. Whether a user exists or not is determined by the password manager.

ObjectMethod eachUser () ->

Twiki::ListIteratorofcUIDs

See baseclass for documentation

ObjectMethod eachGroupMember (\$group) -> listIteratorofcUIDs

See baseclass for documentation

ObjectMethod isGroup (\$user) -> boolean

See baseclass for documentation

ObjectMethod eachGroup () -> ListIteratorofgroupnames

See baseclass for documentation

ObjectMethod eachMembership (\$cUID) -> ListIteratorofgroupsthisuserisin

See baseclass for documentation

ObjectMethod isAdmin (\$cUID) -> \$boolean

True if the user is an admin

- is \$TWiki::cfg{SuperAdminGroup}
- is a member of the \$TWiki::cfg{SuperAdminGroup}

ObjectMethod findUserByEmail (\$email) -> \@cUIDs

- \$email - email address to look up

Return a list of canonical user names for the users that have this email registered with the password manager or the user mapping manager.

The password manager is asked first for whether it maps emails. If it doesn't, then the user mapping manager is asked instead.

ObjectMethod getEmails (\$name) -> @emailAddress

If \$name is a user, return their email addresses. If it is a group, return the addresses of everyone in the group.

The password manager and user mapping manager are both consulted for emails for each user (where they are actually found is implementation defined).

Duplicates are removed from the list.

ObjectMethod setEmails (\$cUID, @emails) -> boolean

Set the email address(es) for the given user. The password manager is tried first, and if it doesn't want to know the user mapping manager is tried.

StaticMethod mapper_getEmails (\$session, \$user)

Only used if passwordManager->isManagingEmails==false (The emails are stored in the user topics.

Note: This method is PUBLIC because it is used by the tools/upgrade_emails.pl script, which needs to kick down to the mapper to retrieve email addresses from TWiki topics.

StaticMethod mapper_setEmails **(\$session, \$user, @emails)**

Only used if `passwordManager->isManagingEmails = false`. (emails are stored in user topics)

ObjectMethod getMustChangePassword (\$cUID) -> **\$flag**

Returns 1 if the \$cUID must change the password, else 0. Returns undef if \$cUID not found.

ObjectMethod getUserData (\$cUID) -> \$dataRef

Return a reference to an array of hashes with user data, used to manage users. Each item is a hash with:

- {name} - name of field, such as "email"
- {title} - title of field, such as "E-mail"
- {value} - value of field, such as "jimmy@example.com"
- {type} - type of field: text, password, checkbox, label
- {size} - size of field, such as 40
- {note} - comment note, if any

User management forms can be build dynamically from this data structure. Each password manager may return a different set of fields.

ObjectMethod setUserData (\$cUID, \$dataRef)

Set the user data of a user. Same array of hashes as `getUserData` is assumed, although only {name} and {value} are used.

ObjectMethod findUserByWikiName (\$wikiname) -> **listofcUIDsassociatedwiththatwikiname**

See baseclass for documentation

The `$skipExistenceCheck` parameter is private to this module, and blocks the standard existence check to avoid reading `.htpasswd` when checking group memberships).

ObjectMethod checkPassword (\$login, \$password) -> **\$boolean**

Finds if the password is valid for the given user.

Returns 1 on success, undef on failure.

ObjectMethod setPassword

(**\$cUID, \$newPassU, \$oldPassU, \$mcp**) -> **\$boolean**

BEWARE: \$user should be a cUID, but is a login when the resetPassword functionality is used. The UserMapper needs to convert either one to a valid login for use by the Password manager

TODO: needs fixing

If the \$oldPassU matches matches the user's password, then it will replace it with \$newPassU.

If \$oldPassU is not correct and not 1, will return 0.

If \$oldPassU is 1, will force the change irrespective of the existing password, adding the user if necessary.

Otherwise returns 1 on success, undef on failure.

ObjectMethod passwordError () -> \$string

returns a string indicating the error that happened in the password handlers TODO: these delayed error's should be replaced with Exceptions.

returns undef if no error

This topic: TWiki > TWikiUsersTWikiUserMappingDotPm

Topic revision: r1 - 2011-08-21 - TWikiContributor



Copyright © 1999-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback

Note: Please contribute updates to this topic on TWiki.org at

TWiki:TWiki.TWikiUsersTWikiUserMappingDotPm.